

Travail de bachelor 2009

Filière Informatique de gestion

Drill on Facebook



Etudiant : André Lambelet

Professeur : Nicole Glassey

PRÉFACE

Dans ce monde où l'on entend de plus en plus parler de réseaux sociaux, le nombre d'étudiants présents sur ce genre de plateformes augmente chaque jour. Un des principaux acteurs dans ce domaine, et sans doute le plus controversé est Facebook. Avec plus de 20% des utilisateurs d'internet qui le visitent quotidiennement, Facebook est le 2^{ème} site internet le plus visité en Suisse après Google¹.

Facebook permet entre autres de créer des applications afin de les mettre à disposition de tout un chacun. Grâce au système d'invitations et à la visibilité sur le profil des membres, la diffusion d'une application est grandement facilitée au niveau mondial. Ainsi, une application simple constituée d'un questionnaire permettant de se comparer à ses amis sur des sujets divers a pu être visitée par plus de 1'300 personnes par jour dès le 11^e jour après son lancement et atteindre plus de 100 pays différents un mois plus tard.

Partant de ce constat, l'intérêt d'une présence dans ce type de média est grand. C'est pourquoi, dans le cadre du projet E-creation 4, Mme Glassey a souhaité développer un drill de programmation à destination des étudiants et disponible sur Facebook ainsi que sur Cyberlearn² (centre E-Learning de la HES-SO).

Page de garde : Image 1 - HES-SO Driller

¹ Source : <http://www.alexa.com/siteinfo/facebook.com> (le 13.07.2009)

² <http://cyberlearn.hes-so.ch>



SOMMAIRE

1	Remerciements.....	4
2	Glossaire	5
3	Présentation du projet	6
4	Description de l'application.....	8
5	Gestion de projet.....	55
6	Satisfaction personnelle.....	60
7	Déclaration sur l'honneur.....	61
8	Bibliographie	62
9	Annexes.....	64
10	Table des illustrations.....	91
11	Table des tableaux.....	92
12	Table des schémas.....	92
13	Table des sources.....	93
14	Table des matières.....	94



1 REMERCIEMENTS

Je tiens à remercier toutes les personnes ayant contribué de près ou de loin à la réalisation de ce travail.

Un merci tout particulier à Mme Nicole Glassey, professeur à la HES-SO Valais et initiatrice de ce projet, pour son appui et ses conseils avisés tout au long de l'élaboration de ce travail de Bachelor.

Je tiens également à remercier :

M. Florian Doche, professeur à la HES-SO Valais, pour ses conseils et orientations lors du développement, M. Hui Duan, assistant à la HES-SO, pour toute la partie de design et développement du jeu Flash, M. Bernard Crettenand, assistant à la HES-SO Valais pour les aspects liés à Moodle, M. David Russo, professeur à la HES-SO Valais, pour la mise en place du serveur.

Un grand merci également aux personnes ayant accepté de participer à la relecture de ce rapport, ainsi qu'à toutes les personnes m'ayant soutenu tout au long de ce travail.

Un langage de programmation est censé être une façon conventionnelle de donner des ordres à un ordinateur. Il n'est pas censé être obscur, bizarre et plein de pièges subtils (ça ce sont les attributs de la magie).

Dave Small, Ingénieur informaticien américain



2 GLOSSAIRE

Amfphp	Format d'échange de données basé sur SOAP (transmission de messages entre objets distants) pour PHP
API	Application Programming Interface. Ensemble de classes définissant les moyens de recourir aux services d'un fournisseur
Cyberlearn	Centre E-learning de la HES-SO basé sur Moodle
DMZ	Demilitarized Zone. Sous-réseau accessible depuis l'extérieur (internet)
HES-SO Driller	Surnom du projet
HESSODriller	Nom du serveur applicatif de ce projet
HTML	HyperText Markup Language
iFrame	Cadre permettant d'afficher un document HTML externe au milieu d'une page web
Adresse IP	Adresse personnelle d'un ordinateur connecté à internet
iPhone	Téléphone mobile d'Apple
JavaDoc	Outil permettant de créer une documentation d'API en format HTML selon les commentaires présents dans un code source Java
JavaScript	Langage de programmation de scripts pour pages web
Moodle	Plateforme d'apprentissage en ligne
MySQL	Système de gestion de base de données
PHP	Langage de programmation web
Serveur applicatif	Serveur sur lequel sont installées des applications utilisées à distance par les clients
Service Web (WebService)	Programme fournissant une fonctionnalité particulière à d'autres programmes sur internet
XML	Langage de balisage générique utilisé pour le stockage ou transfert de données
XSD	Langage permettant de définir la structure d'un document XML
XSLT	Langage de transformation XML (par exemple vers de l'HTML)



3 PRÉSENTATION DU PROJET

3.1 CONTEXTE

Initié par Mme Glassey, le projet original "Drill on Facebook" comprend deux membres (Florian Doche et Frédéric Bapst) ainsi qu'une personne en charge du Développement (Hui Duan). Dans le cadre de ce Travail de Bachelor ce projet est appelé HES-SO Driller afin de bien faire la distinction avec le projet original.

HES-SO Driller a pour but de motiver les étudiants lors de l'apprentissage de la programmation de base par le côté ludique du jeu mais également le côté compétitif et communautaire de Facebook. Le contexte innovant ainsi que la collaboration avec deux plateformes déjà connues des étudiants devrait permettre de favoriser l'utilisation de ce drill afin de pousser le public cible à la pratique régulière des exercices proposés, tout en apportant une visibilité plus importante à la HES-SO auprès des étudiants ou futurs étudiants présents sur Facebook.

3.2 DÉLIVRABLES

Le rendu du travail de Bachelor est composé de trois parties distinctes :

- l'application
- le rapport final
- la présentation de défense du projet

Livrée sur le CD annexé, l'application est détaillée et expliquée dans ce rapport final.



3.3 CAHIER DES CHARGES

Afin de définir les spécificités requises pour l'application, un cahier des charges (voir annexes) a été établi d'entente entre l'auteur et la responsable du projet et accepté lors de la séance hebdomadaire du 26 juin 2009.

Points principaux :

- HES-SO Driller sera composé d'une partie jeu et une partie administration
- Le jeu sera disponible sur Facebook et Moodle
- Il devra proposer les points suivants :
 - Possibilité pour les professeurs de créer et personnaliser un cours
 - Blocage des accès à un cours par un mot de passe
 - Quatre types de quizz prédéfinis à choix lors de la création
 - Gestion des scores et comparaisons aux autres utilisateurs
 - Statistiques des résultats obtenus par tous les étudiants
 - Possibilité pour les étudiants de proposer ou commenter une question



4 DESCRIPTION DE L'APPLICATION

4.1 INTRODUCTION

Dans cette partie principale, nous allons passer en revue l'architecture du projet, du côté physique et applicatif. Une documentation technique répertoriant et détaillant toutes les spécificités du développement ainsi qu'un manuel d'utilisation de chacune des deux parties seront également proposés.

4.2 ARCHITECTURE PHYSIQUE

La configuration physique de ce projet est relativement simple. Les interactions sont, quant à elles, sensiblement plus complexes.

4.2.1 ACTEURS PRINCIPAUX

Trois acteurs principaux sont présents:

- Serveur applicatif
- Serveur Moodle
- Serveur Facebook

4.2.2 SERVEUR APPLICATIF

Le serveur principal HESSODriller est constitué d'une machine virtuelle installée avec Windows Server 2003. L'application devant être disponible depuis l'extérieur du réseau de l'HES-SO Valais, HESSODriller a dû être configuré en DMZ par le service informatique. De manière à accéder aux données de l'application depuis le Flash, un service web a été créé sur le serveur applicatif.

4.2.3 SERVEUR MOODLE

Un service web supplémentaire a été mis en place sur le serveur de Moodle afin de récupérer les informations de profil des visiteurs car la base de données y relative n'est disponible qu'en lecture et ce uniquement en interne.



4.2.4 VUE GLOBALE

Voici une vue globale des différents acteurs et leurs interactions :

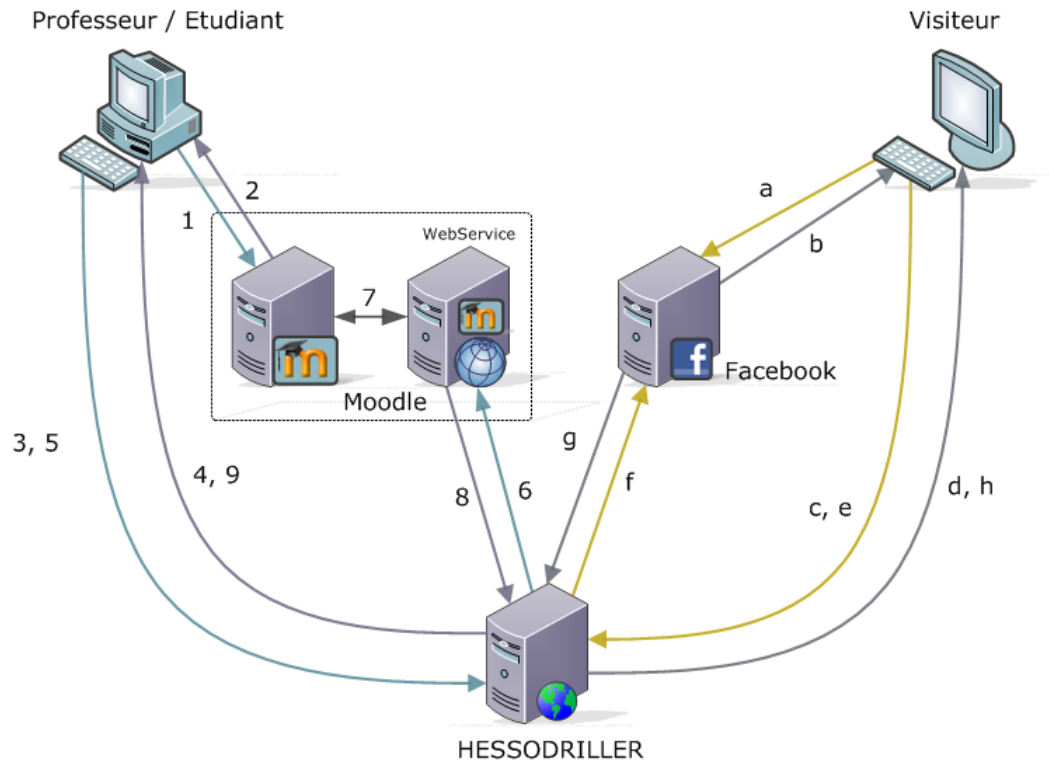


Image 2 - Architecture physique

Légende :

- | | |
|---|---|
| 1. Authentification et requête de la page web | a. Requête de la page web |
| 2. Retour de la page HTML au navigateur | b. Retour de la page HTML au navigateur |
| 3. Requête de l'iframe au serveur HESSODriller | c. Requête de l'iframe au serveur HESSODriller |
| 4. Retour de l'iframe | d. Retour de l'iframe |
| 5. Requête du flash et appels au WebService HESSODriller | e. Requête du Flash et appels au WebService HESSODriller |
| 6. Appels au WebService Moodle | f. Appels à l'API Facebook |
| 7. Interaction avec la base de données Moodle | g. Retours de l'API |
| 8. Retours du WebService Moodle | h. Retour du Flash et des appels au WebService HESSODriller |
| 9. Retour du Flash et retours des appels au WebService HESSODriller | |



4.3 ARCHITECTURE APPLICATIVE

Ce projet devant être intégré à Facebook et à Moodle, le choix de la technologie s'est très rapidement orienté vers l'utilisation du PHP comme langage de programmation principal car l'API développée par Facebook est principalement prévue pour ce langage. Le choix de la base de données s'est donc effectué en conséquence et s'est porté sur MySQL.

L'architecture 5-tiers suivante a été développée et mise en place :

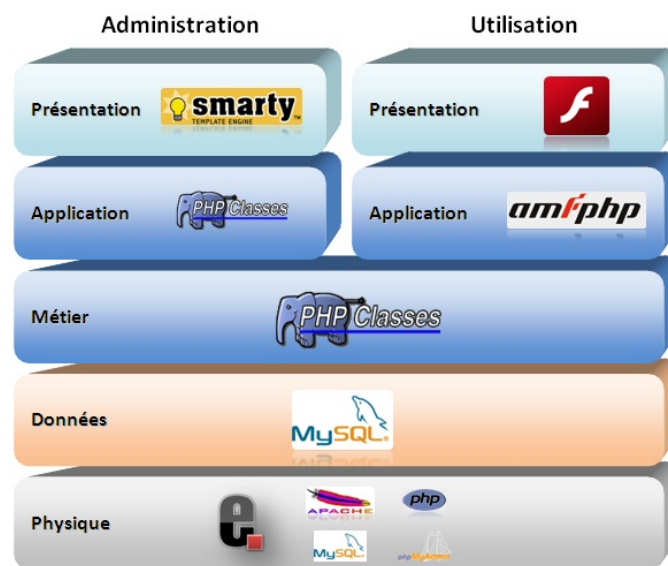


Image 3 - Architecture 5-tiers



4.3.1 COUCHE PHYSIQUE

Le serveur utilisé lors du développement est EasyPhp³. Ce dernier étant un serveur de développement "tout-en-un", une solution plus poussée devra être mise en place lors du passage en production afin de supporter la charge nécessaire.

4.3.2 COUCHE DONNÉES

Le serveur de base de données utilisé est MySQL. La base de données décrite au chapitre « Modélisation », a été implémentée avec le moteur de stockage InnoDB car il permet de définir des contraintes d'intégrité référentielles grâce à l'utilisation des clefs étrangères.

4.3.3 COUCHE MÉTIER

Les divers objets métiers ont été développés selon un schéma prédéfini. Pour chaque table de la base de données une classe a été créée avec les méthodes suivantes :

- constructeur
- chargement des objets persistants de la base de données
- ajout
- modification
- suppression

D'autres méthodes ont bien entendu été ajoutées en fonction des besoins.

4.3.4 COUCHE APPLICATION

La couche application est formée des pages PHP où est effectué tout le traitement des données pour la partie administration.

Pour la partie utilisation, cette couche est représentée par les classes « Remote » ainsi que par l'implémentation d'Amfphp qui, en utilisant la technologie de sérialisation, permet une communication directe entre PHP et Flash.

³ <http://www.easyphp.org>



4.3.5 COUCHE PRÉSENTATION

En ce qui concerne la partie jeu, toute la couche présentation est faite en Flash.

Pour l'administration, un moteur de template a été utilisé : Smarty⁴. Son utilisation permet de s'affranchir totalement de tout lien entre le design et le code. Cela permet de développer ces deux aspects de manière totalement asynchrone. Smarty permet surtout de créer un nouveau design sans avoir à toucher le code applicatif pour pouvoir, par exemple, proposer un affichage sur ordinateur, téléphone mobile ou iPhone en ne changeant que la référence au fichier de design utilisé.

⁴ <http://www.smarty.net>



4.4 INFORMATIONS TECHNIQUES

4.4.1 INTRODUCTION

Dans cette section se trouvent toutes les informations concernant aussi bien les choix effectués que les outils utilisés ainsi qu'une documentation du code et des problèmes rencontrés lors du développement.

4.4.2 CHOIX DE L'ARCHITECTURE LOGICIELLE

4.4.2.1 INTÉGRATION DE FLASH À FACEBOOK

Selon l'annexe « Choix d'Architecture Flash - Facebook », la technologie d'intégration de notre application au sein de Facebook s'est portée sur une iFrame. Ainsi, moins d'adaptations sont nécessaires lors du passage de l'application destinée à Moodle à celle intégrée à Facebook.

4.4.3 MODÉLISATION

En fonction de l'analyse effectuée ainsi que des informations qu'il est possible de récupérer depuis Moodle ou Facebook, un schéma de données a été établi et vous est présenté maintenant. A la suite du schéma complet vous trouverez une description détaillée de chaque partie du modèle.



4.4.3.1 MODÈLE COMPLET

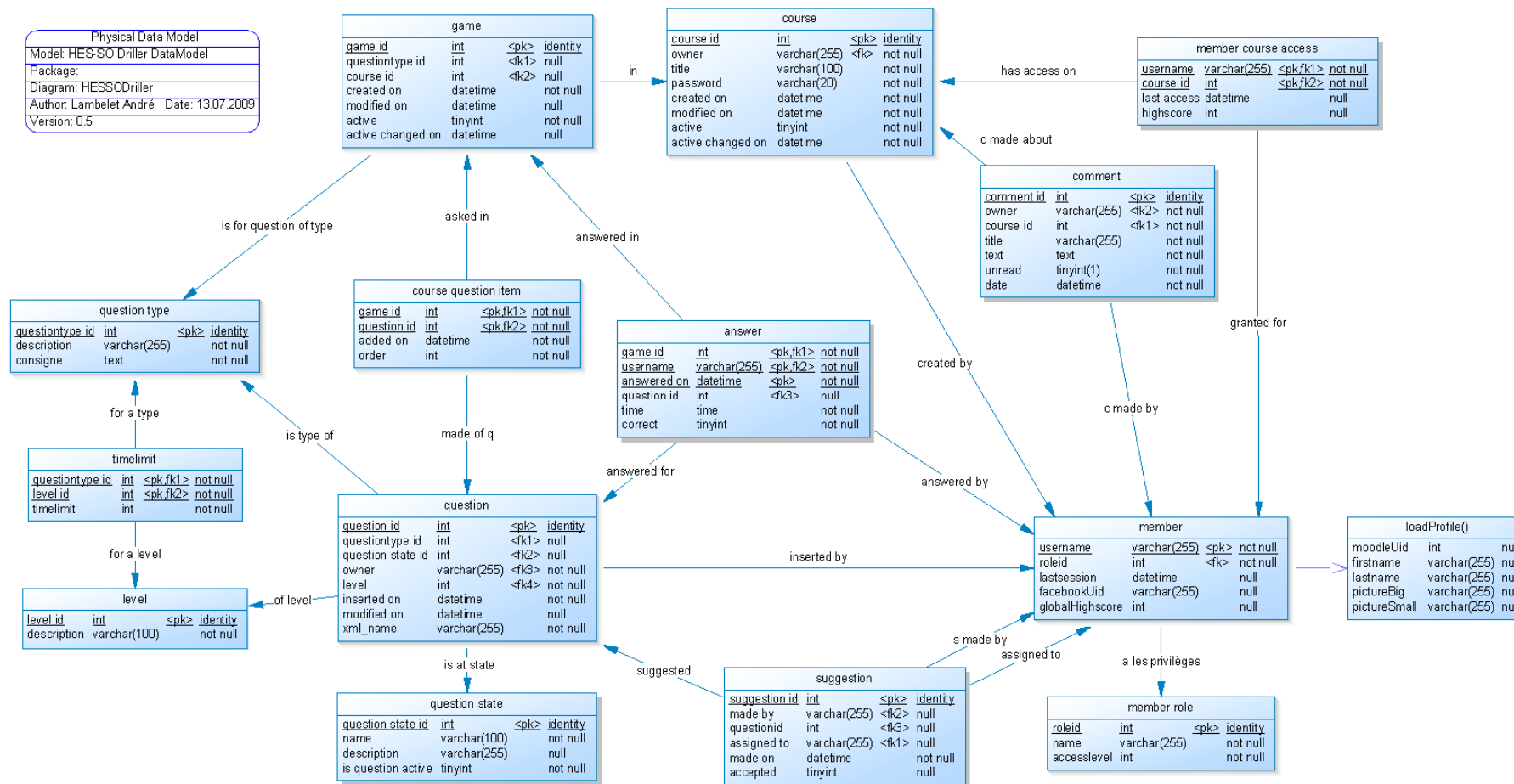


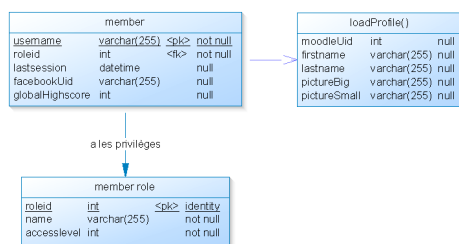
Schéma 1 - Modèle physique de la base de données



4.4.3.2 DESCRIPTION

MEMBRES

La gestion des membres se fait par le nom d'utilisateur Moodle des visiteurs. Pour les visiteurs de Facebook, le *username* est le *userId* Facebook avec le préfixe "facebook_". Il n'est pas nécessaire de sauvegarder les données tels que le nom, mot de passe ou photo des utilisateurs car ces données sont accessibles par le Webservice Moodle ou par l'API Facebook. La définition des droits d'accès est gérée grâce aux rôles attribués.



La table portant le nom *loadProfile()* est présente à titre informatif afin de définir les nouvelles données présentes dans l'objet *Member* une fois le chargement des données depuis le Webservice Moodle effectué avec succès.

Schéma 2 - BDD Membres

COURS

Chaque cours peut contenir un nombre indéterminé de jeux. Chaque jeu est défini pour un type de question spécifique. L'administrateur aura alors la possibilité d'ajouter autant de questions que nécessaire dans un jeu. Les réponses sont liées au jeu afin de pouvoir les distinguer dans le cas où elles seraient présentes dans deux jeux différents.

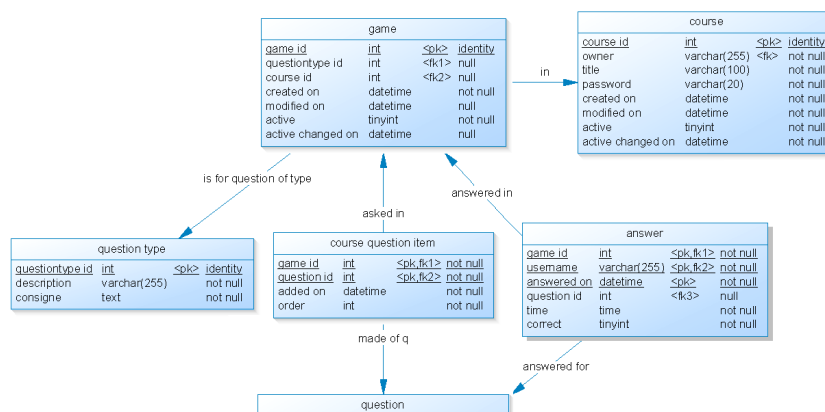


Schéma 3 - BDD Cours et Jeux



QUESTIONS

Les attributs de chaque question ainsi que le chemin d'accès du fichier XML du contenu sont stockés dans la base de données avec une référence à l'état de la question. Cet état spécifie si la question est disponible ou non au grand public. Cela permet d'ajouter des questions disponibles uniquement dans les cours configurés par les professeurs.

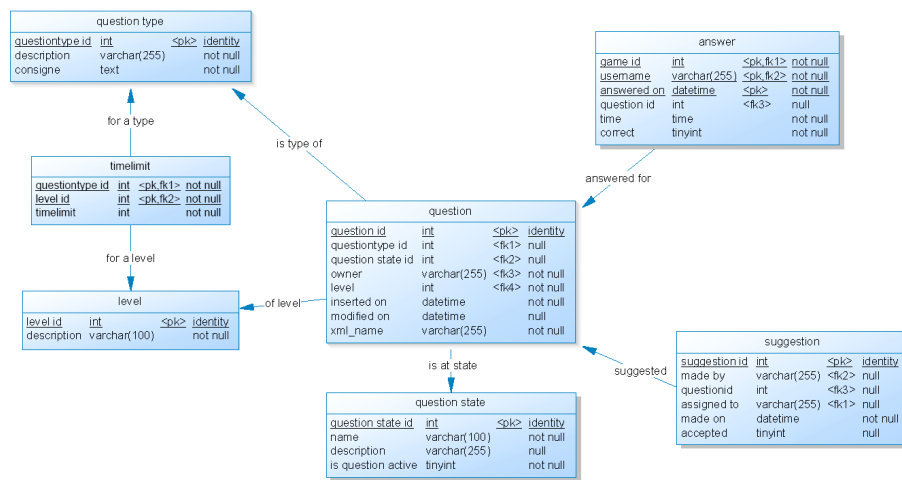


Schéma 4 - BDD Questions

Cette gestion des questions est agrémentée de plusieurs tables permettant de gérer les différents niveaux de difficulté ainsi que le temps de réponse accordé en fonction du type de question. D'autre part, les réponses sont stockées afin de pouvoir extraire n'importe quelle statistique désirée.

Lors de l'ajout d'une suggestion, une nouvelle entrée dans la table *question* est créée et référencée dans la table *suggestion* afin de sauvegarder les informations quant à l'utilisateur qui l'a envoyée.



ACCÈS AUX COURS ET COMMENTAIRES

Les accès aux cours sont stockés dans la base de données afin de ne pas redemander le mot de passe lors de chaque visite. Cela permet également de stocker les meilleurs scores par étudiant pour chaque cours.

D'autre part, la possibilité de donner un commentaire n'est proposée aux utilisateurs que pour un cours, c'est pourquoi une relation a été faite avec la table *course*.

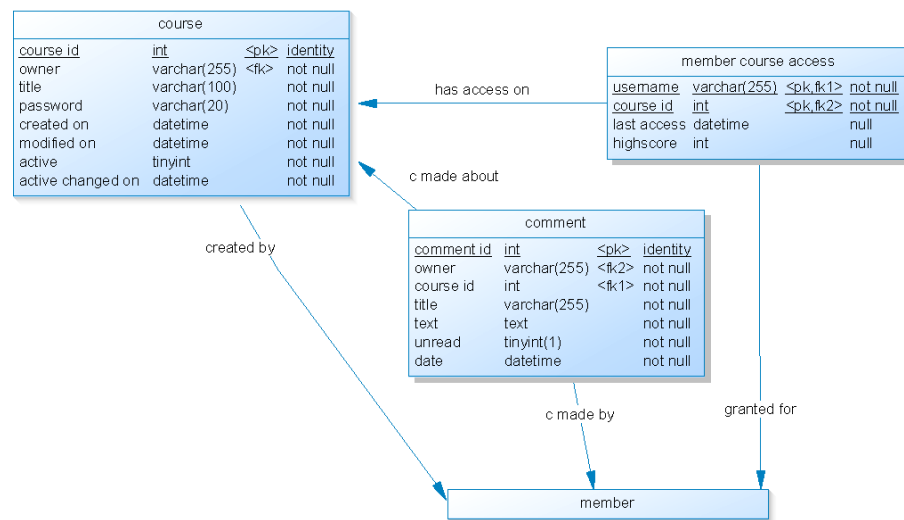


Schéma 5 - Accès aux cours et commentaires



4.4.4 RÉSULTATS DE L'ANALYSE

Cette section regroupe toutes les décisions prises en accord avec la responsable du projet, durant la période d'analyse du projet.

4.4.4.1 NAVIGATION DANS LE JEU

Les différentes possibilités de navigation au sein de l'application pour un visiteur ont été définies selon le schéma suivant :

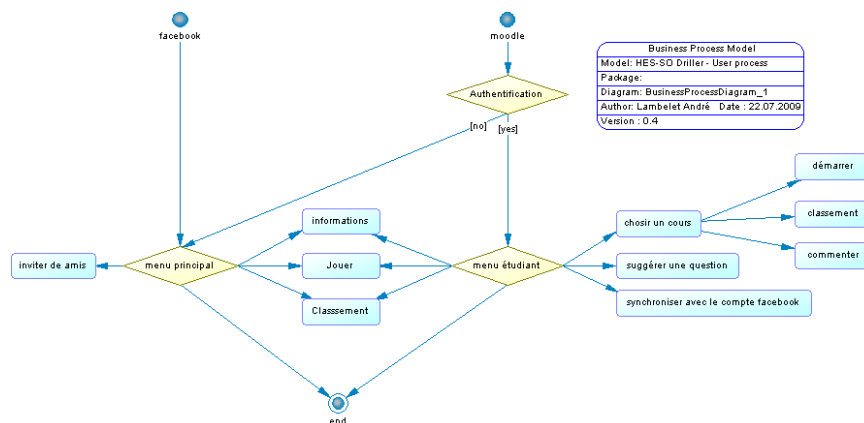


Schéma 6 - Navigation dans l'application

Le jeu en lui-même se déroulera de la manière suivante :

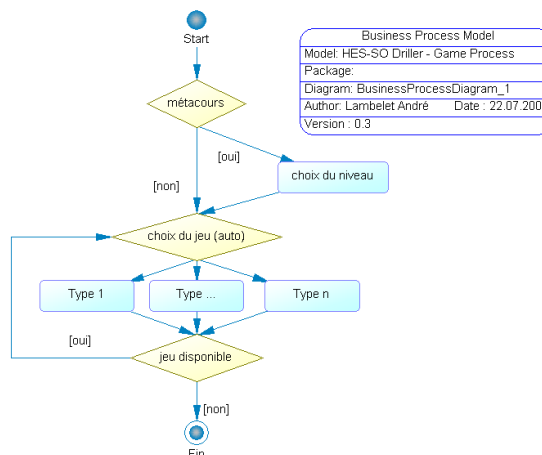


Schéma 7 - Déroulement du jeu



4.4.4.2 TYPES DE QUESTIONS

Quatre types de questions ont été définis :

- Question à 4 réponses
- Complétez le code
- Trouvez l'erreur
- Réorganisez les lignes de code

Le premier type sera obligatoirement implémenté tandis que les autres le seront en fonction du temps disponible.

QUESTION À 4 RÉPONSES

Ce premier type est constitué d'une simple donnée ainsi que quatre réponses au maximum. Il est possible d'avoir plusieurs réponses justes mais également de n'avoir que deux ou trois réponses possibles, selon les besoins du professeur.

Ce type de jeu donne la possibilité de poser des questions de type "texte à trous", ceci simplement en utilisant le caractère de soulignement "_" à l'endroit désiré.

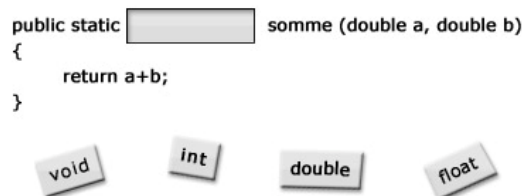


Image 4 – Type de question « Texte à trous »



COMPLÉTEZ LE CODE

Quasiment similaire au précédant, ce type propose un algorithme incomplet et quatre possibilités de complétion dont une seule permet d'obtenir le résultat affiché.

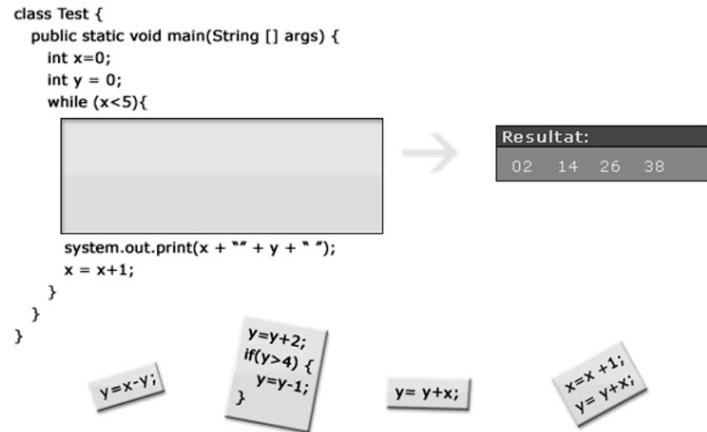


Image 5 - Type de question « Complétez le code »

TROUVEZ L'ERREUR

Afin de tester la rapidité d'assimilation du code par les étudiants, ce type sera proposé de manière simple. Un code sera affiché ligne par ligne à l'écran et l'étudiant devra cliquer sur la ligne comportant une erreur.

RÉORGANISEZ LES LIGNES DE CODE

Troisième type envisagé, la réorganisation des lignes de codes se présentera comme suit : des lignes d'un code sources seront affichées dans le désordre, l'étudiant devant alors réorganiser le code de manière à obtenir un résultat donné et valider son choix.



Image 6 - Type de question
« Réorganisez les lignes de code »



4.4.4.3 CONTENU DU MÉTA-COURS

Le méta-cours proposés aux visiteurs de Facebook ainsi qu'aux étudiants sera composé de dix questions par type de jeu, choisies aléatoirement au sein de toutes les questions disponibles.

4.4.4.4 PROTOTYPES ÉCRANS

Afin de guider le designer Flash, des prototypes écrans (disponibles en annexes) ont été créés afin de prévoir tous les éléments nécessaires sur les différentes interfaces du jeu.

4.4.4.5 MÉTHODES DU WEBSERVICE

Les différentes méthodes requises par le Flash ont été définies d'un commun accord en spécifiant la classe, la signature de chaque méthode ainsi que le type de retour. Vous en trouverez la liste en annexe.

4.4.4.6 CALCUL DU SCORE

Sachant que le temps effectif est égal à zéro si la réponse est fausse, le procédé de calcul du score a été défini comme suit:

$$\text{Score total} = \left(\sum^{\text{questions}} \text{Temps maximum} - \text{Temps effectif} \right) \times \left(1 + \frac{\text{niveau}}{10} \right)$$

Ainsi le score augmente si l'élève répond rapidement et de manière correcte. Il est également favorisé s'il choisit des questions de niveau supérieur.



4.4.5 CLASSES EXTERNES

Plusieurs classes externes ont été implémentées dans ce projet. Que ce soit pour faire face à des problèmes rencontrés ou pour simplifier certaines parties du code, vous en trouverez ci-dessous une courte description.

4.4.5.1 AMFPHP

Afin de pouvoir accéder de manière simple et efficace aux données de l'application depuis Flash, l'implémentation d'Amfphp a été grandement bénéfique. En effet, l'implémentation de cet outil open-source est aisée. Après avoir copié les sources nécessaires et configuré le répertoire contenant les classes à mettre à disposition la configuration est terminée.

Amfphp propose également un service de test des différentes classes et méthodes à disposition. Cet outil nous a été très utile dans le cadre de l'élaboration des méthodes du service web afin de connaître le résultat obtenu par l'appel des différentes méthodes.



Image 7 - Logo amfphp

Cette librairie nous a également donné la possibilité de mettre à disposition les objets PHP pour leur utilisation dans Flash. Ainsi, nous avons pu utiliser les classes métier préalablement créées et donc bénéficier d'un gain de temps considérable.

4.4.5.2 EZSQL

Afin de gérer les connexions à la base de données MySQL, une classe externe a été utilisée afin de simplifier l'utilisation des appels à la base de données. Avec ses fonctions de débogage, ezSQL a grandement simplifié la gestion des accès à la base de données.



Image 8 - Logo ezSQL



4.4.5.3 API FACEBOOK

L'intégration d'une application iFrame à Facebook se fait par l'intermédiaire de l'API Facebook. Cet API est mise à disposition du public gratuitement et permet d'appeler les différentes fonctions mises à disposition. Cependant, il faut être conscient que les données privées des membres ne sont pas aisément consultables dès que l'on sort du cercle d'amis de la personne connectée. Ceci a posé passablement de problèmes lors du calcul des résultats par exemple. L'aide étant néanmoins relativement complète, toutes les informations nécessaires ont pu être trouvées facilement. La mise en place a, quant à elle, nécessité plus de temps à cause de problèmes liés aux sessions. Vous trouverez plus d'informations sur les problèmes rencontrés dans le chapitre « Analyse et explication des problèmes rencontrés »



Image 9 - Logo Facebook

4.4.5.4 FACEBOOKBRIDGE

Afin d'interagir avec l'API Facebook depuis Flash, cette classe permet de faire appel à n'importe quelle méthode de l'API et de récupérer le résultat sans problème. Cette solution développée par Zerofractal studio⁵ a été très facilement mis en place.



Image 10 - Logo FacebookBridge

4.4.5.5 FILEUPLOAD

Originellement destinée à l'envoi d'images, cette classe a été intégrée à ce projet afin de simplifier la gestion des erreurs d'envoi des fichiers XML et XSD. Elle pourra cependant être utilisée dans le cas où une amélioration de l'application nécessiterait l'envoi et la gestion d'images.

⁵ <http://www.zerofractal.com>



4.4.5.6 HTML2XML

Ce JavaScript permet, en utilisant la nomenclature des éléments `<form>` et `<input>` de la source HTML, de générer un fichier XML possédant les attributs désirés et le contenu inséré par l'utilisateur.

4.4.5.7 SMARTY

Décrite dans le chapitre « Couche Présentation » cette librairie permet de scinder le code en une partie traitement et une partie affichage, permettant de changer aisément le design sans toucher au traitement des données. Cet outil est pratique lorsque plusieurs personnes s'occupent séparément du code et du design.



Image 11 - Logo Smarty

4.4.5.8 TINYMCE

Afin de permettre la mise en forme des données des questions, la mise en forme d'un éditeur de texte intégré à la page de modification a été entreprise. Reprenant les contrôles standards des éditeurs de texte grand public, cet éditeur permet à tout un chacun de mettre en forme ses questions sans difficulté.



Image 12 - Logo TinyMCE

4.4.5.9 XMLFORM

Cette librairie permet de générer un formulaire HTML en fonction d'un fichier de configuration XML décrivant les données ainsi qu'une feuille de style XSLT afin de formater la sortie. Cette solution a été utilisée afin de donner la possibilité aux administrateurs de l'application d'ajouter facilement de nouveaux types de questions.

4.4.5.10 XSLT-PHP4-TO-PHP5

Afin de pouvoir utiliser la librairie XMLForm prévue pour la version 4 de PHP, une simple inclusion de la classe `xslt-php5-to-php5` a permis la transition des méthodes désormais dépréciées dans la nouvelle version installée sur notre serveur (PHP 5).




4.5 SPÉCIFICITÉS DE L'IMPLEMENTATION

Afin de permettre une meilleure compréhension du code et algorithmes développés, voici les différentes spécificités mises en œuvre.

4.5.1 NIVEAUX D'ACCÈS

Les trois niveaux d'accès suivants ont été définis de manière arbitraire. Chaque type d'accès bénéficie de droits étendus par rapport au niveau inférieur. Voici les principales caractéristiques des accès.



20 Administrateur	<ul style="list-style-type: none"> • Gestion des types de questions • Gestion des droits d'accès
10 Professeur	<ul style="list-style-type: none"> • Gestion des questions • Gestion des cours
1 Utilisateur	<ul style="list-style-type: none"> • Participer au jeu • Commenter un cours • Proposer une question

Tableau 1 - Droits d'accès

Afin de pouvoir, si nécessaire, ajouter d'autres niveaux intermédiaires, un écart a été prévu entre les niveaux actuels. Ainsi nous pourrions ajouter un niveau 5 pour des professeurs pouvant uniquement éditer des cours et non ajouter de nouvelles questions, ou un niveau 15 pour des administrateurs ne pouvant que gérer les types de questions sans pouvoir modifier les droits d'accès des utilisateurs.



4.5.2 WEB SERVICE MOODLE

La base de données Moodle n'étant pas accessible depuis l'extérieur, un espace web sur le serveur de tests donnant l'opportunité d'y déposer des fichiers PHP ainsi qu'un accès en lecture à la base de données a gracieusement été mis à disposition par M. Crettenand, responsable du système Moodle.

Au moyen de cet accès, la mise en place d'un système d'authentification ainsi que d'un service d'obtention des informations du profil a pu être mis en place. Regroupés en un seul fichier, voici une description de ces deux services.

4.5.2.1 AUTHENTIFICATION

Initialement prévu, un service web a été mis en place afin de valider le couple login – mot de passe des utilisateurs avec la base de données de Moodle car, selon le responsable du système, il n'aurait pas été possible de récupérer les informations de la session de l'utilisateur authentifié sur Moodle.

Ce service est actuellement constitué d'un fichier PHP déposé sur le serveur de test de Cyberlearn.

Il reçoit deux paramètres qui sont le nom d'utilisateur et le mot de passe crypté. Après comparaison avec la base de données MySQL de Moodle, le service retourne le résultat de l'authentification de l'utilisateur, de type booléen.

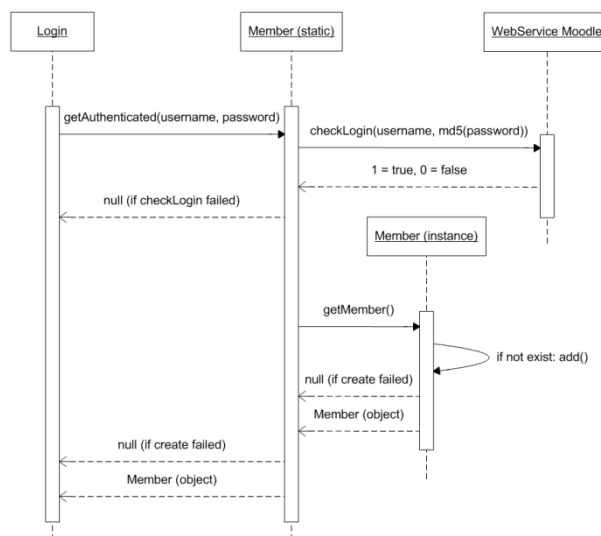


Schéma 8 - Diagramme de séquence de l'authentification



Après de plus amples recherches, Hui Duan a trouvé une solution permettant, avec certains paramètres qu'il est possible de passer entre Moodle et la page incluse dans l'iFrame, de récupérer le nom d'utilisateur de la session Moodle. Ceci étant sécurisé grâce à l'encodage de l'adresse IP du client couplé à une clef privée de Moodle. Nous avons donc pu implémenter un nouveau type d'authentification offrant l'avantage de ne pas revalider les informations d'accès des utilisateurs de Moodle.

```
function identification($accesslevel = 1) {
    $log = null;
    $mdp = null;

    if(isset($_SERVER["HTTP_REFERER"]))
    (
        // si l'on accède bien depuis moodle et que l'adresse ip du visiteur est identique à celle de référence sur moodle
        if (isset($_GET["host"]) &&
            strpos(isset($_SERVER["HTTP_REFERER"]) ? $_SERVER["HTTP_REFERER"] : null, $_GET["host"]) !== false &&
            md5($_SERVER["REMOTE_ADDR"].SECRET_CL) == isset($_GET['cc']) ? $_GET['cc'] : null)
        {
            $log = (isset($_GET['uid']) ? $_GET['uid'] : '');
            $mdp = md5(SECRET_CL);
        }
    )
    ...
}
```

Source 1 - Sécurité de l'identification venant de Moodle

Les conditions imposées sont les suivantes :

- La page n'est pas accédée directement
- La page de référence correspond à une page hébergée sur le serveur Moodle
- L'adresse IP du visiteur n'a pas changé entre le login sur Moodle et l'accès à la page actuelle.

La sécurité de ce système est estimée suffisante par rapport à la sensibilité des données hébergées.

4.5.2.2 RÉCUPÉRATION DU PROFIL

Le jeu développé présentant un classement des participants, il a fallu récupérer les informations de l'utilisateur afin de pouvoir afficher d'autres informations que simplement le nom d'utilisateur. Afin d'obtenir le nom, prénom ainsi que la photo, un deuxième service a été mis en place. Il fournit les informations nécessaires en fonction du nom d'utilisateur Moodle fourni.



4.5.3 TYPES DE QUESTIONS DYNAMIQUES

Afin de permettre l'évolution de l'application et d'ajouter de nouveaux types de questions, le contenu de celles-ci est sauvegardé au format XML. L'extensibilité va encore plus loin car le formulaire d'ajout d'une question a été dynamisé par la génération automatique des champs nécessaires. Utilisant la librairie XMLForm, une feuille de style XSLT permet la synthèse du code HTML en fonction du schéma XML décrivant la structure des questions.

Le script HTML2XML permet ensuite de générer dynamiquement le code XML de la question en fonction des éléments des formulaires présents sur la page. Ce code est ensuite validé par un fichier XSD qui contient la structure finale des questions voulues.

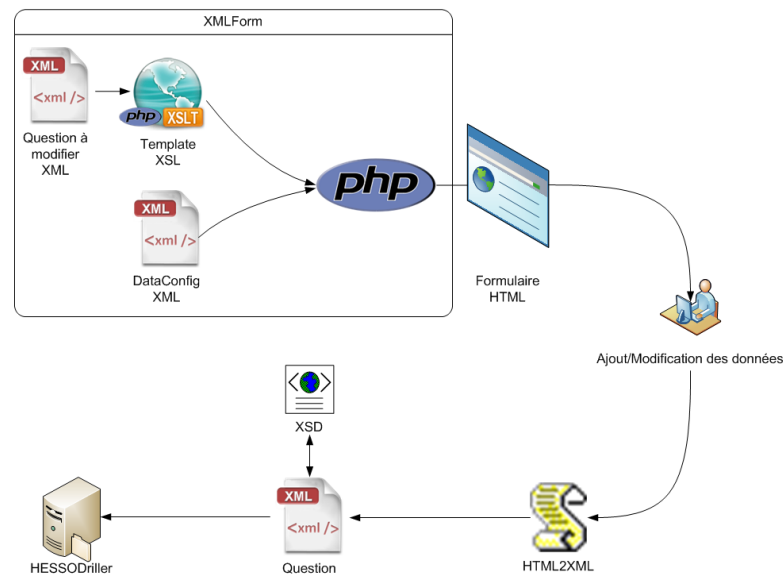


Schéma 9 - Types de questions dynamiques

Cela donne la possibilité aux administrateurs de n'avoir qu'à créer un nouveau fichier XML de configuration de la question ainsi qu'un schéma XSD de validation afin d'ajouter un nouveau type de question. Il faudra bien entendu créer une nouvelle classe Flash qui sera utilisée pour l'affichage des questions de ce nouveau type.

Vous trouverez plus de détails techniques de cette implémentation dans le chapitre « Analyse et explication des problèmes rencontrés ».



4.5.4 GESTION DE L'ÉTAT DES QUESTIONS

Actuellement les deux états de disponibilité d'une question sont :

- Disponible
- Non disponible

A noter que ceci influence uniquement la disponibilité des questions pour le méta-cours. Il est dans tous les cas possible pour les professeurs d'inclure cette question dans un cours.

La structure est par contre prévue pour intégrer un nombre indéfini d'états différents. Nous pourrions donc, par exemple, avoir l'utilité de deux états intermédiaires :

- Disponible uniquement pour Moodle
- Disponible uniquement pour Facebook

Afin de mettre en œuvre ces deux états supplémentaires il suffirait de rajouter deux lignes dans la table *question_state* de la base de données (mises en évidence sur l'image suivante) et d'adapter les classes *Remote* afin de gérer ces nouveaux cas.

QUESTION_STATE_ID	NAME	DESCRIPTION	IS_QUESTION_ACTIVE
1	Active	Disponible	1
2	Désactivée	Non Disponible	0
3	Active (Moodle)	Disponible uniquement sur Moodle	0
4	Active (Facebook)	Disponible uniquement sur Facebook	1

Image 13 - Ajout d'états supplémentaires



4.5.5 FLASH

Les sources du prototype fonctionnant partiellement sur Moodle, transmises par le designer Flash, intégraient les fonctionnalités suivantes:

- Jeu global avec choix du niveau
- Choix du cours avec contrôle de l'accès
- Affichage du classement des utilisateurs
- Gestion des questions de type "question à 4 réponses"
- Affichage des différents types de questions d'un cours
- Navigation dans les menus
- Ecran d'information sur l'application
- Envoyer un commentaire
- Lien "Ajouter une question"

A partir de ces sources, plusieurs modifications y ont été apportées, notamment :

- Correction des bugs dans la navigation
- Ajout d'un bouton de synchronisation avec le compte Facebook
- Gestion de plusieurs types de question par jeu : une fois le cours démarré, l'utilisateur voit la liste des types compris dans le jeu, le prochain à effectuer étant en surbrillance. Il n'a toutefois pas la possibilité de choisir l'ordre dans lequel il passe les quiz.
- Gestion d'un nouveau type de question, "trouvez l'erreur" : Afin d'intégrer un nouveau type de question, une modification de la structure des classes a dû être effectuée. C'est pourquoi une nouvelle classe a été créée afin de gérer un nouveau type de réponse possible. L'algorithme contient quand à lui un déroulement différent, adapté en fonction du type.

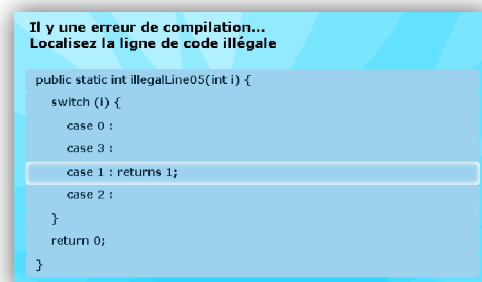


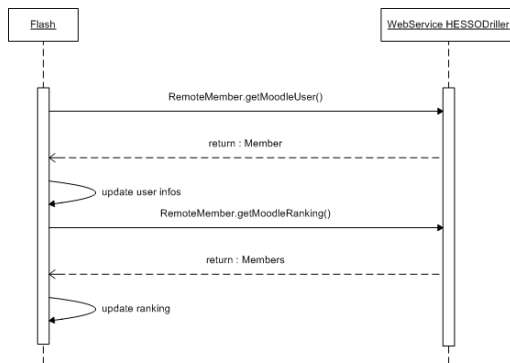
Image 14 - Type "trouvez l'erreur"



L'adaptation du prototype afin d'être intégré à Facebook a également été entreprise.

Comme il n'était pas possible d'obtenir les informations de l'utilisateur par le service web, il a fallu modifier toutes les procédures d'obtention des informations.

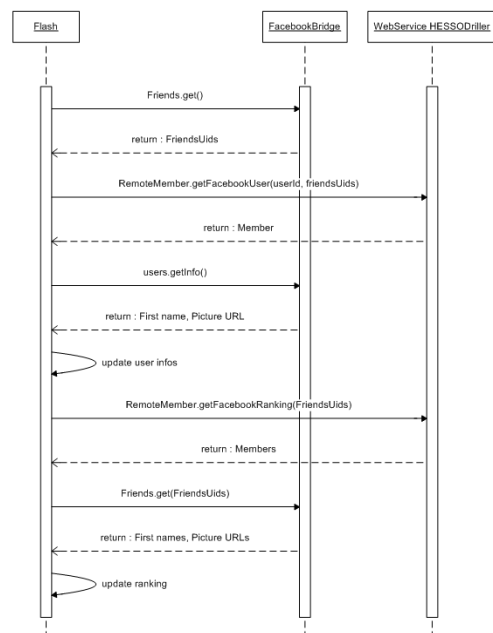
Moodle



Le service web retournant toutes les informations, la mise à jour du profil de l'utilisateur ainsi que du classement peut se faire directement au retour des appels.

Schéma 10 - Flash (Moodle), obtention des informations des utilisateurs

Facebook



Dans ce cas, afin d'obtenir les informations du profil ainsi que du classement des amis, il est d'abord nécessaire d'avoir la liste exhaustive des amis pour que le service web retourne uniquement les membres amis de l'utilisateur.

Une fois ces informations obtenues il est nécessaire de faire encore appel à Facebook afin d'obtenir les noms et photos des utilisateurs.

Schéma 11 - Flash (Facebook), obtention des informations des utilisateurs



Afin de faire participer un maximum de personnes il est commun, sur Facebook, d'inviter ses amis à ajouter une application. Pour se faire une page a été spécialement créée. Le bouton permettant d'y accéder a bien entendu été ajouté au Flash.



Image 15 - Bouton "Inviter des amis"

Quelques problèmes ont été rencontrés lors de la création de cette page. Vous en trouverez les détails dans la section y relative.

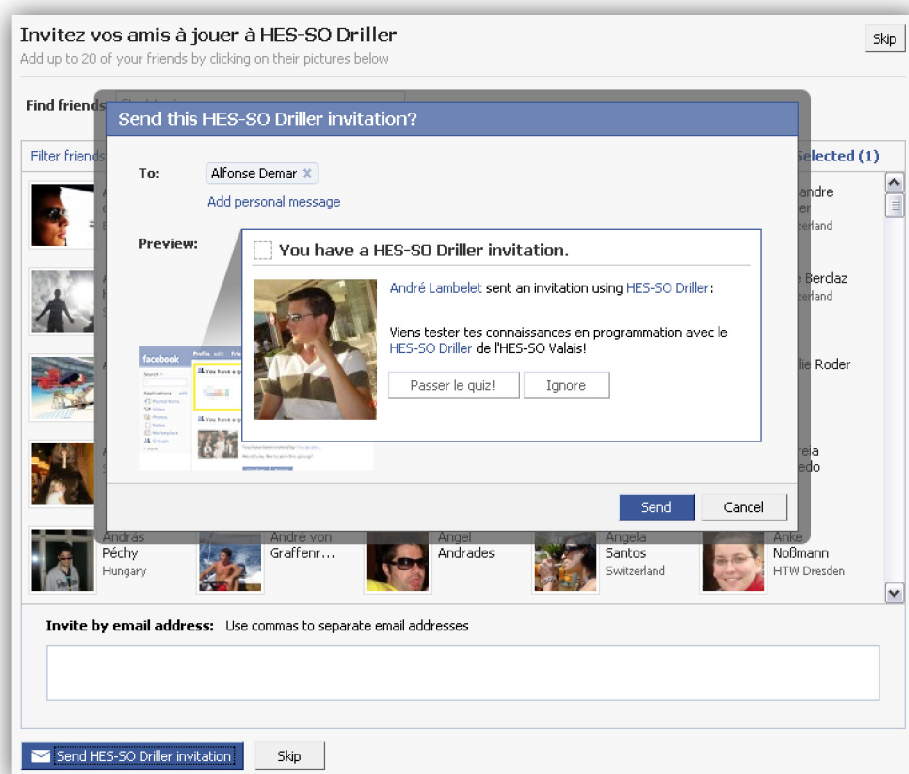


Image 16 - Formulaire d'invitation d'amis



4.6 ANALYSE ET EXPLICATION DES PROBLÈMES RENCONTRÉS

Au cours de ce projet, plusieurs problèmes ont été rencontrés. Voici une explication de la manière dont ces derniers ont été résolus, avec une classification selon 3 niveaux : faible (1), moyen (2), élevé(3).

4.6.1 TYPES DE QUESTION DYNAMIQUES

Ce problème a été estimé comme étant de niveau 3 (élevé).

4.6.1.1 PROBLÉMATIQUE

Il nous a fallu trouver une solution afin de pouvoir créer une interface d'administration des questions dynamiques. En effet, l'application étant prévue pour accueillir plusieurs types de questions fondamentalement différents et surtout modifiable en tout temps, il n'était donc pas possible de prévoir toutes les possibilités de champs possibles pour une question.

La problématique a pu être divisée en deux phases :

1. Générer un formulaire HTML dynamique en fonction des besoins.
2. Créer un document XML à partir d'un formulaire HTML.
3. Permettre la modification aisée des questions

4.6.1.2 RÉOLUTION

1. Grâce à la librairie XMLForm, une solution a finalement été trouvée afin de générer un formulaire HTML de manière dynamique en fonction d'un fichier de configuration XML décrivant les données ainsi qu'une feuille de style XSLT afin de formater la sortie. Afin de pouvoir implémenter avec succès la librairie XMLForm il a d'abord fallu activer

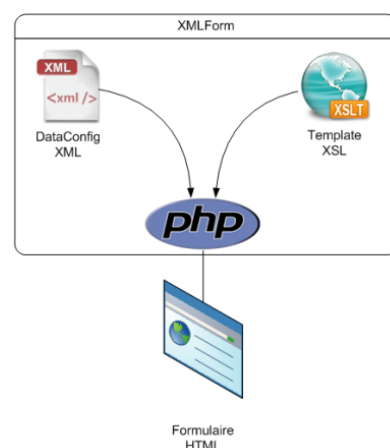


Schéma 12 - Génération du formulaire



l'extension *php.xsl.dll*. Malheureusement cette librairie était prévue pour une version de PHP antérieure à celle installée sur notre serveur. Pour faire face à cette incompatibilité, la classe *xslt-php5-to-php5* a été utilisée de manière à gérer la transition entre les versions 4 et 5 de PHP sans devoir modifier l'intégralité des classes de la librairie.

```
<xsl:template name="form-element">
  <xsl:choose>

    <xsl:when test="@type='donnee'"> <!-- Donnée-->
      <form>
        <xsl:attribute name="name"><xsl:value-of select="@name" /></xsl:attribute>
        <textarea>
          <xsl:attribute name="name">text</xsl:attribute>
          <xsl:attribute name="value">text</xsl:attribute>
          <xsl:for-each select="html-display-option">
            <xsl:attribute name="{@name}"><xsl:value-of select="@value"/></xsl:attribute>
          </xsl:for-each>
        </textarea>
      </form>
    </xsl:when>

    * * *
```

Source 2 - Définition de l'élément *donnee* (Template.xsl)

```
<form-config>
  <var name="donnee" type="donnee">
    <description>Donnée</description>
    <html-display-option name="class" value="mceAdvanced"/>
    <html-display-option name="style" value="width:100%"/>
    <html-display-option name="size" value="100"/>
    <html-display-option name="maxlength" value="25"/>
  </var>

  <var name="reponse1" type="reponse">
    <description>Réponse 1</description>
  </var>
  <var name="reponse2" type="reponse">
    <description>Réponse 2</description>
  </var>
  <var name="reponse3" type="reponse">
    <description>Réponse 3</description>
  </var>
  <var name="reponse4" type="reponse">
    <description>Réponse 4</description>
  </var>
</form-config>
```

Source 3 - Configuration du type "question à 4 réponses" (XML)



- Une fois les champs remplis par l'administrateur, le JavaScript HTML2XML nous a permis de générer un arbre XML contenant tous les champs générés de la question que l'on a pu enregistrer dans un fichier après l'avoir validé avec le XSD afin de vérifier que tous les champs soient correctement remplis et que la génération du XML ait été effectuée correctement.

```

<?xml version="1.0" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Question" type="QuestionType" />
  <xs:complexType name="QuestionType">
    <xs:sequence>
      <xs:element name="donnee" type="donneeType" />
      <xs:element name="reponse" type="reponseType"/>
      <xs:element name="reponse" type="reponseType"/>
      <xs:element name="reponse" type="reponseType"/>
      <xs:element name="reponse" type="reponseType"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="donneeType">
    <xs:sequence>
      <xs:element name="text" type="xs:string" />
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="reponseType">
    <xs:sequence>
      <xs:element name="text" type="xs:string" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="name" type="xs:string" />
    <xs:attribute name="right" type="xs:boolean" />
  </xs:complexType>
</xs:schema>

```

Source 4 - Schéma de validation du type "question à 4 réponses" (XSD)

```

$dom = new DOMDocument();
$dom->loadXML($xml);
//mise à jour
if ($questionId != null)
{
  $questionType = Question_Type::getSingleQuestion_Type($questionTypeId);
  // calcul les millisecondes actuelles (http://benjisimon.blogspot.com/2007/09/currenttimemillis-in-php.html)
  list($usec, $sec) = explode(" ", microtime());
  $xmlFilename = "QuestionType-{$questionType->id}_" . $user . "-" . round(((float)$usec + (float)$sec * 1000), ".xml");

  if (!$dom->schemaValidate(SCHEMAS_PATH . "QuestionType-{$questionType->id}_.xsd"))
  {
    array_push($errors, "Merci de remplir tous les champs");
    $action = 'add';
  }
  else
  {
    ...
  }
}

```

Chargement du schéma XML à valider

Définition du nom du futur fichier contenant la question

Validation du schéma

Source 5 - Validation de la question



3. Afin de pouvoir également modifier une question sans qu'un professeur ne doive connaître la syntaxe XML, il a fallu trouver une astuce afin d'intégrer les données dans le formulaire HTML généré au point 1. Nous avons donc généré le fichier XSL par un fichier PHP afin de charger le fichier XML adapté en fonction de la question à modifier.

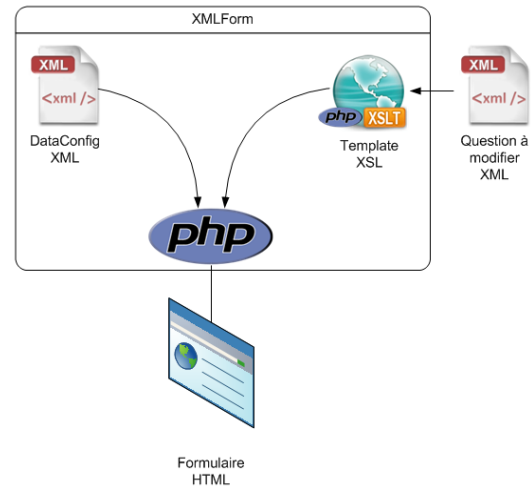


Schéma 13 - Génération du formulaire de modification

```

<?php
header ("content-type: text/xml");
$datas = isset($_REQUEST['cf']) ? $_REQUEST['cf'] : null;
$loadDatas = $datas ? '<xsl:variable name="datas" select="document(' . urlencode($datas) . ')"/>' : "";

?>
...
<xsl:template name="form-element">
<xsl:choose>

<xsl:when test="@type='donnee'"> <!-- Donnée-->
<?php echo $loadDatas ?>
<form>
<xsl:attribute name="name"><xsl:value-of select="@name" /></xsl:attribute>
<textarea>
<xsl:attribute name="name">text</xsl:attribute>
<xsl:attribute name="value">text</xsl:attribute>
<xsl:for-each select="html-display-option">
<xsl:attribute name="{@name}"><xsl:value-of select="@value"/></xsl:attribute>
</xsl:for-each>
<xsl:value-of select="$datas/Question/donnee/text"/>
</textarea>
</form>
</xsl:when>
...

```

Définition de l'adresse du contenu de la question à modifier

Chargement du fichier

Source 6 - Génération du Template.xml par PHP



4.6.2 SESSIONS FACEBOOK

Ce problème a été estimé comme étant de niveau 3 (élevé).

4.6.2.1 PROBLÉMATIQUE

L'utilisation de l'API Facebook utilise les références de la session de l'utilisateur actuellement connecté à Facebook dans le navigateur. Ceci nous a posé de gros problèmes au niveau du service web.

En effet, le service web devant calculer tous les classements et résultats des utilisateurs en fonction de la personne connectée, il fallait savoir qui étaient ses amis et récupérer leurs informations. Malheureusement un problème s'est rapidement posé lors de l'élaboration de ces algorithmes car apparemment l'utilisation d'*amfphp* empêchait la récupération des informations de sessions.

4.6.2.2 RÉOLUTION

Après de longues heures de recherche, aucune solution n'a été trouvée au niveau du service web. Il a donc été décidé, en accord avec le designer Flash, que toutes les informations personnelles ainsi que les listes d'amis de l'utilisateur courant seraient récupérées directement depuis Flash. Ceci étant possible selon les tests effectués durant l'analyse. Les méthodes du service web ont donc été modifiées en conséquence.



4.6.3 TINYMCE & HTML2XML

Ce problème a été estimé comme étant de niveau 2 (moyen).

4.6.3.1 PROBLÉMATIQUE

Durant l'implémentation de TinyMCE nous avons constaté qu'il n'était plus possible au script HTML2XML de récupérer le contenu mis en forme par cet éditeur de texte. Le XML généré ne contenant alors plus le contenu ni les balises adéquates.

4.6.3.2 RÉOLUTION

Après avoir recherché les raisons de l'absence du contenu il est apparu que le texte modifié par l'éditeur est effectivement intégré par le script au formulaire uniquement au moment de l'envoi de ce dernier.

Il a donc fallu éditer la méthode de création du XML par le script HTML2XML pour intégrer un appel à l'objet *TinyMCE* afin de lui faire enregistrer le contenu avant la création du fichier XML et non après.

```
function computeXML(theForm) {  
    tinyMCE.activeEditor.save();  
    var elm="";  
    var formX = document.forms[0];  
    var nbe=0;  
    for (i=0; i<formX.elements.length; i++) {  
        if (formX.elements[i].name=="XMLVersion")  
            var txt = "<" + formX.XMLVersion.value + ">\n";  
        else if (formX.elements[i].name=="element")  
            nbe = nbe + 1;  
    }  
    elm = new Array(nbe-1);  
    var j=0;  
    for (i=0; i<formX.elements.length; i++) {  
        if (formX.elements[i].name=="element") {  
            elm[j++] = formX.elements[i].value;  
            txt = txt + "<" + formX.elements[i].value + ">\n";  
        }  
    }  
    ...  
}
```

Source 7 - HTML2XML



4.6.4 FORMULAIRE D'INVITATION D'AMIS

Ce problème a été estimé comme étant de niveau 1 (faible).

4.6.4.1 PROBLÉMATIQUE

Afin d'utiliser les contrôles fournis par Facebook afin d'inviter des amis (*<fb:multi-friend-selector>*) il était, selon la documentation fournie, judicieux d'utiliser la technologie *fbconnect* qui permet, grâce à un JavaScript, d'utiliser les contrôles spéciaux de Facebook à l'intérieur d'une iFrame externe. Après plusieurs tests, aucune solution n'a été trouvée.

4.6.4.2 RÉOLUTION

La solution utilisée a été de forcer l'intégration de la page d'invitation par *FBML* afin de ne pas devoir utiliser de JavaScript après le retour de la page HTML mais que les contrôles soient directement générés par Facebook avant de renvoyer la page au visiteur. La page d'invitation est alors simplement constituée du code suivant:

```

1 <fb:fbml>
2   <fb:request-form
3     action="index.php"
4     method="POST"
5     invite="true"
6     type="HES-SO Driller"
7     content="Viens tester tes connaissances en programmation avec le <fb:application-name /> de
1  HES-SO Valais!<fb:req-choice url='http://apps.facebook.com/hessodriller' label='Passer le quiz!' />"
8   <fb:multi-friend-selector
9     showborder="false"
10    actiontext="Invitez vos amis à jouer à HES-SO Driller">
11  </fb:request-form>
12 </fb:fbml>

```

Source 8 - Page d'invitation d'amis



4.7 AMÉLIORATIONS PROPOSÉES

4.7.1 INTERACTIONS AVEC FACEBOOK

Dans le cadre de l'utilisation sur Facebook, plusieurs améliorations sont envisageables. En effet, il est commun sur cette plateforme d'ajouter des interactions entre les amis afin de créer une fidélisation des utilisateurs. Voici quelques pistes à explorer :

- Publier les nouveaux records d'un utilisateur sur son profil
- Notifier les amis d'un utilisateur lorsque celui-ci leur passe devant dans le classement
- Déplacer le formulaire d'invitation d'amis à un endroit plus stratégique. Par exemple avant de pouvoir afficher son score. Attention, ceci pourrait être considéré comme de l'incitation abusive. Les utilisateurs ne doivent pas être obligés d'inviter des amis pour continuer
- Ajouter un classement global de tous les utilisateurs (pas seulement les amis)
- Ajouter un emplacement dans les "encarts" du profil du joueur (emplacement destiné aux applications, leur permettant de publier des informations statiques)
- Proposer aux utilisateurs de défier des amis

4.7.2 STATISTIQUES DES RÉPONSES DES ÉTUDIANTS

Afin de donner un feedback aux professeurs concernant leurs quiz, il serait intéressant de leur proposer des statistiques des réponses données par les étudiants. Ainsi il serait possible d'afficher pour chaque cours:

- Nombre d'élèves ayant répondu
- Score moyen d'un élève
- Moyenne générale des meilleurs scores
- Temps de réponse au cours complet, par jeu ou par question
- Toute autre statistique désirée...



4.7.3 SÉCURITÉ

La gestion du système d'authentification unique depuis Moodle fonctionne sans problème avec le navigateur Firefox ou lorsque la page est intégrée au wrapper de Moodle comme c'est le cas pour l'administration de la plateforme.

Un problème subsiste par contre lorsqu'un étudiant accède au jeu avec Internet Explorer. En effet, ce dernier, pour des raisons de sécurité, ne fournit plus la variable serveur *HTTP_REFERER*. Cette variable étant utilisée afin de vérifier que personne n'ait changé les paramètres entre le clic sur le lien dans Moodle et l'accès au Flash, il a donc dû modifier l'algorithme de vérification des informations. En voici la version modifiée :

```
$moodleOK = false;
if (isset($_GET["host"]) &&
    strpos(isset($_SERVER["HTTP_REFERER"]) ? $_SERVER["HTTP_REFERER"] : null, $_GET["host"]) !== false &&
    md5($_SERVER["REMOTE_ADDR"].SECRET_CL) == isset($_GET['cc']) ? $_GET['cc'] : null)
{
    $moodleOK = true;
}
if (isset($_SERVER["HTTP_REFERER"]) &&
    md5($_SERVER["REMOTE_ADDR"].SECRET_CL) == isset($_GET['cc']) ? $_GET['cc'] : null)
{
    $moodleOK = true;
}
```

Source 9 - Algorithme de SSO modifié

Cette version permet malheureusement à un utilisateur de changer le nom d'utilisateur dans l'adresse et d'accéder au jeu en tant que quelqu'un d'autre. Bien que la seule conséquence de ce trou de sécurité soit la sauvegarde d'un nouveau score plus élevé ou des réponses données à un cours, cela pourrait être judicieux de trouver une parade à ce problème dans le cas où les données des quiz seraient utilisées à des fins d'évaluation par exemple.



4.7.4 GESTION DES DROITS AUTOMATIQUES

Toujours dans le registre de l'authentification des utilisateurs, il pourrait être pratique d'automatiser la partie d'attribution des droits "professeur" aux utilisateurs.

Actuellement, un administrateur doit donner les droits à chaque professeur individuellement. Cela veut donc dire qu'ils ne peuvent pas accéder à l'administration tant que l'administrateur ne les a pas promus.

En ajoutant un nouveau service web sur Moodle retournant le statut de l'utilisateur par rapport au cours Moodle "Drill on Facebook" il serait possible d'attribuer automatiquement les droits de niveau 5 aux administrateurs du cours lors de leur authentification sur la plateforme.

La partie à modifier se situe dans la méthode *Member::getMember(\$username)*, ligne 196.

```
if (!$resultset)
{
    $newUser = new Member($username, 1, null);
    $newUser->add();
    if (Member::getMember($username) == false)
        return null;
}
```

Source 10 - Méthode getMember() originale

Serait à remplacer par :

```
if (!$resultset)
{
    $role = getRoleFromMoodleWebService($username);
    $newUser = new Member($username, $role, null);
    $newUser->add();
    if (Member::getMember($username) == false)
        return null;
}
```

Source 11 - Méthode getMember() modifiée

Après avoir, bien entendu, créé la méthode d'interrogation du service web Moodle, ici nommée *getRoleFromMoodleWebService()*.



4.8 DOCUMENTATION

4.8.1 INTRODUCTION

Afin de faciliter la mise en place de cette application ainsi que la prise en main de la partie administration de la plateforme, voici deux guides spécifiques, une courte explication quant à la structure des fichiers ainsi qu'une note sur la documentation du code.

4.8.2 DÉPLOIEMENT

4.8.2.1 SPÉCIFICATIONS MINIMALES DU SERVEUR

Le serveur doit être installé avec au minimum les éléments standards suivants :

- Apache 2.2.11
- PHP 5.2.8
- MySQL 5.0.51a

Concernant la configuration PHP il faut, de plus, activer l'extension *php.xml.dll* afin de pouvoir utiliser la librairie *XMLForm*.

Le matériel utilisé sera également important car l'application Facebook devrait normalement se répandre relativement vite. Il devra donc supporter la charge de plusieurs milliers d'utilisateurs journaliers. Les requêtes de sauvegarde des réponses ont été optimisées afin d'éviter de faire appel trop souvent à la base de données mais il serait judicieux de concevoir une architecture matérielle avec un serveur web et un serveur MySQL séparé.



4.8.2.2 INSTALLATION

HESSODRILLER

Une fois le serveur web installé et configuré selon les spécificités du serveur définies plus haut, il suffit de copier les sources PHP disponibles sur le CD (répertoire *www*) et charger le fichier de création de la base de données *HESSODriller.sql* également disponible dans le répertoire *www*.

WEBSERVICE MOODLE

L'installation du service web Moodle se fait en envoyant le fichier *webservice.php* de la racine du serveur web sur le serveur de Moodle, en modifiant si besoin les informations d'accès à la base de données.

4.8.2.3 CONFIGURATION

Dans les exemples suivants, l'adresse du serveur web est *http://153.109.124.66/*. Il faut bien entendu l'adapter en fonction de votre configuration.

MOODLE

En ce qui concerne la configuration de Moodle il est possible de configurer l'application depuis n'importe quel cours désiré. Pour se faire il faut ajouter deux ressources de type "liens vers un fichier ou site web" et les configurer comme suit.

Administration :

- Emplacement : *http://153.109.124.66/*
- Fenêtre : Même fenêtre
- Cocher "afficher la page de navigation sur la même page"
- Paramètres :

uid	Utilisateur - Nom d'utilisateur ▼
host	Divers - URL du serveur (http://cyberlearn.hes-so.ch) ▼
cc	Divers - Code chiffré ▼

Image 17 - Paramètres de configuration Moodle



Jeu :

- Emplacement : *http://153.109.124.66/flash/index.php*
- Fenêtre : Nouvelle fenêtre
- Désélectionner : toutes les cases à cocher suivantes
- Largeur de la fenêtre par défaut : 740
- Hauteur de la fenêtre par défaut : 680
- Paramètres : idem partie Administration

FACEBOOK

La configuration de l'application sur Facebook se déroule de la manière suivante : après avoir ajouté la nouvelle application il faut paramétrer les points suivants:

Paramètres	Valeur
Canvas Page URL	À choix
Canvas Callback URL	<i>http://153.109.124.66/flash/index.php?host=fb</i>
Render Method	IFrame
IFrame Size	Resizable
Canvas Width	Full width (760px)
Quick Transitions	On
Connect URL	<i>http://153.109.124.66/fbconnect</i>
Account Preview URL	<i>http://153.109.124.66/</i>
Application Type	Web

Tableau 2 - Configuration Facebook



HESSODRILLER

Une fois l'application Facebook configurée, il faut éditer le fichier *config.php* situé dans le répertoire *inc* du serveur web et éditer les points suivants:

Nom de la variable statique	Valeur
MAIN_URL	http://153.109.124.66/
MOODLE_URL	Répertoire contenant le fichier <i>webservice.php</i> sur le serveur Moodle
ROOT_PATH	Adresse locale du répertoire web (ex : <i>c:/www/</i>)
MYSQL_USER	Selon configuration MySQL
MYSQL_PASS	Selon configuration MySQL
MYSQL_DB	Selon configuration MySQL
MYSQL_HOST	Selon configuration MySQL
FB_API_KEY	Donnée par Facebook
FB_SECRET	Donnée par Facebook

Tableau 3 - Configuration des variables

Le reste de la configuration dépendant de ces différentes valeurs, il n'est pas nécessaire de modifier d'autres paramètres. Vous avez cependant la possibilité de modifier le titre principal du site web ou le nombre de questions tirées au hasard pour le public. En définissant la variable *LOCAL* à 1 cela active automatiquement toutes les fonctions de débogage utiles lors du développement.



4.8.3 ADMINISTRATION

Destinée aux professeurs ainsi qu'aux administrateurs, la partie d'administration de l'application comporte les menus suivants:



Image 18 - Menu administration

Les menus "Types" et "Droits" ne sont disponibles que pour les administrateurs

4.8.3.1 QUESTIONS

La gestion des questions est divisée en deux parties distinctes.

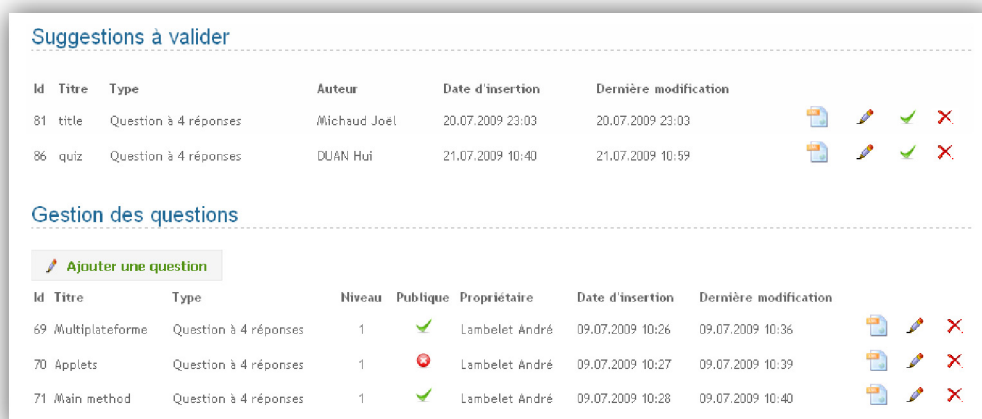


Image 19 - Administration des questions

Dans la première partie s'affichent les suggestions que le professeur a reçues des étudiants. Il est possible alors de consulter le fichier XML de la question, d'éditer la suggestion, de la valider afin de l'intégrer au pool de questions ou simplement de la supprimer.

La deuxième partie répertorie toutes les questions que le professeur a ajouté ou reçues et validées.



4.8.3.2 COURS

Les professeurs n'ont à disposition que les cours qu'ils ont eux-mêmes créés tandis que les administrateurs eux ont tous les cours à disposition.

Gestion des cours

[Ajouter un cours](#)

Número	Titre	Disponible	Jeux	Propriétaire	Mot de passe	Commentaires	Date de création	Dernière modification		
1	/My basic java course	✓	2	Lambelet André	*****	3	01.06.2009 00:00	09.07.2009 10:54		
6	Quick'n'Easy	✓	1	Lambelet André	n/a	1	09.07.2009 11:06	09.07.2009 11:06		
7	Implements	✗	1	Lambelet André	n/a	0	09.07.2009 11:06	09.07.2009 11:06		
9	Find the error	✓	1	Lambelet André	n/a	2	25.07.2009 20:08	25.07.2009 20:08		

Image 20 - Gestion des cours

Sur la première page, il est possible d'activer un cours en cliquant sur ou pour les désactiver. Un accès direct aux commentaires est également possible en cliquant sur le nombre affiché. L'édition d'un cours est possible en cliquant sur le crayon.

L'interface de modification propose plusieurs choix: éditer le titre et mot de passe, activer ou désactiver le cours, modifier ou supprimer un jeu existant, en ajouter un, ou visualiser les commentaires.

Gestion du cours : My basic java course

Número du cours: 1

Titre: My basic java course

Mot de passe: *****

Disponible:

Jeux:

Id	Nom	Questions		
1	Question à 4 réponses	9		
25	Trouvez l'erreur	1		

[Ajouter un jeu](#)

Propriétaire: Lambelet André

Commentaires: [3](#)

Date de création: 01.06.2009 00:00

Dernière modification: 09.07.2009 10:54

Image 21 - Edition d'un cours



4.8.3.3 JEU

My basic java course | Gestion du jeu (Question à 4 réponses)

 Ajouter une question

Id	Titre	Type	Propriétaire	Date d'insertion	Dernière modification	
69	Multiplateforme	Question à 4 réponses	Lambelet André	09.07.2009 10:26	09.07.2009 10:36	 
70	Applets	Question à 4 réponses	Lambelet André	09.07.2009 10:27	09.07.2009 10:39	 
72	Applet Class	Question à 4 réponses	Lambelet André	09.07.2009 10:30	16.07.2009 08:16	 

Image 22 - Gestion d'un jeu

Lors de l'ajout d'un jeu, le type de question désiré est d'abord demandé, puis le professeur arrive sur l'interface d'ajout/suppression des questions du jeu. Il est alors possible de supprimer une question du cours (ce qui ne la supprimera pas de l'application bien entendu) ou en ajouter de nouvelles. Pour l'ajout de nouvelles questions il suffit de cocher les questions désirées dans la liste et valider.

Choix de questions à ajouter : Question à 4 réponses

Selection	Titre	Type	Propriétaire	Date d'insertion	Dernière modification	
<input type="checkbox"/>	Main method	Question à 4 réponses	allstar	09.07.2009 10:28	09.07.2009 10:40	
<input type="checkbox"/>	Layout	Question à 4 réponses	allstar	09.07.2009 10:35	09.07.2009 10:35	
<input type="checkbox"/>	Servlets	Question à 4 réponses	allstar	09.07.2009 10:35	09.07.2009 10:41	

Image 23 - Ajout de questions à un jeu

4.8.3.4 COMMENTAIRES

La gestion des commentaires est aisée. La possibilité est donnée au professeur de marquer les éléments lus ou de les effacer.

Visualisation des commentaires : My basic java course

Lu	Titre	Texte	Propriétaire	Date	
	/My title 1	This is my first comment!	allstar	02.06.2007 16:24	
	Mon commentaire	Test WebService	allstar	01.07.2009 09:08	

Image 24 - Gestion des commentaires



4.8.3.5 TYPES

La gestion des types de question est présentée selon le même principe que pour les éléments précédent. Lors de la modification du type sont demandés les champs basiques. Quant à la configuration de la question ainsi que du schéma de validation, il est demandé d'envoyer directement le fichier car l'édition de ce type de document est bien plus simple avec un éditeur spécialisé plutôt que dans un éditeur en ligne.

Image 25 - Modifier un type de question

Trois limites de temps sont demandées car il est possible de les configurer individuellement. Il a cependant été décidé de donner un temps unique pour tous les niveaux.

4.8.3.6 DROITS

La gestion des droits permet aux administrateurs de gérer les niveaux d'accès de manière simplifiée. Il suffit alors de cliquer sur l'icône de la colonne désirée afin de donner ou non les droits d'accès correspondants. A noter que, pour promouvoir un administrateur, il faut qu'il soit préalablement professeur. D'autre part, il est impossible de supprimer les droits "professeur" à un utilisateur s'il est administrateur.

	Nom	Prénom	Username	Prof.	Admin
	Lambelet	André	allstar	✓	✓
	DUAN	Hui	duanhui	✓	✗
	Crettenand	Bernard	bcettenand	✗	✗
	Glassey	Nicole	nicole.glassey@hevs.ch	✓	✓

Image 26 - Gestion des droits



4.8.4 UTILISATION

Afin de découvrir ce jeu, vous pouvez y accéder par les moyens suivants:



Moodle

Nom du cours	Tests Drill on Facebook
Adresse	http://cyberlearn.hes-so.ch/course/view.php?id=1909
Mot de passe	welcome09



Facebook

Nom du cours	HES-SO Driller
Adresse	http://apps.facebook.com/hessodriller/

L'utilisation du jeu se voulant intuitive, il n'est pas nécessaire d'expliquer en détails la navigation dans le jeu. Vous trouverez néanmoins quelques captures d'écrans ci-dessous afin de découvrir le jeu dans le cas où un accès à l'application en ligne n'est pas possible.

4.8.4.1 CAPTURES D'ÉCRAN

Moodle



Facebook



Choix du cours (Moodle) :



Choix du niveau (tous) :



Contenu du cours (tous) :



Exemple de question (tous) :



Commenter un cours (Moodle) :



Panneau d'information (tous) :



4.8.5 DÉVELOPPEMENT

4.8.5.1 STRUCTURE

Les éléments principaux de la structure de fichiers mise en place sur le serveur web sont représentés sur le schéma suivant :

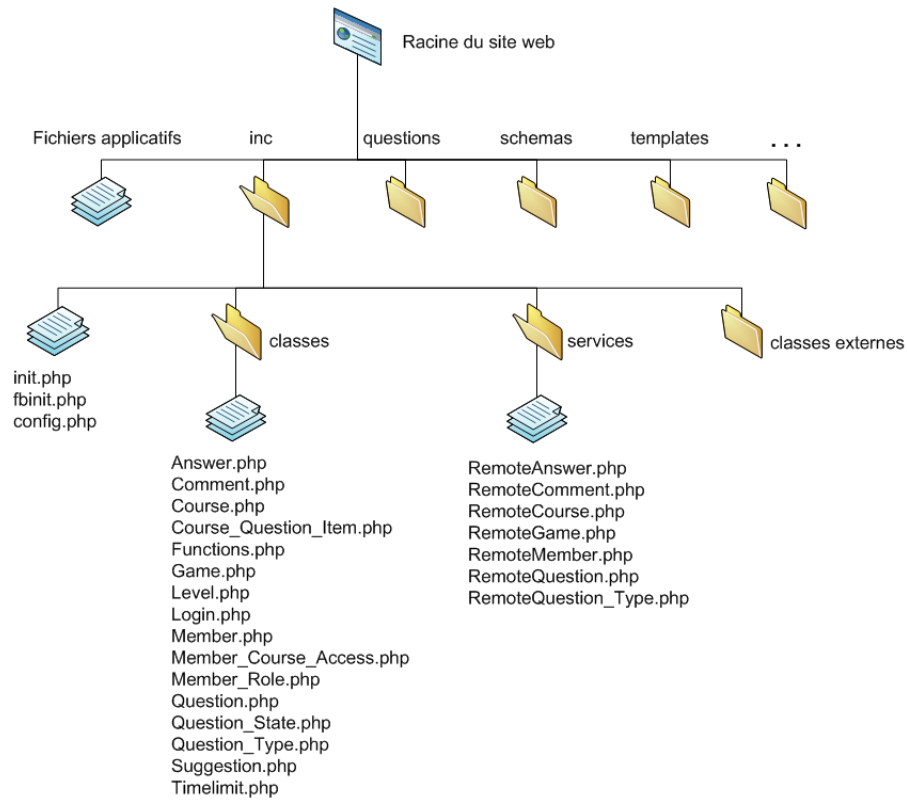


Schéma 14 - Structure des fichiers

Les répertoires *questions* et *schémas* contiennent respectivement les données des questions et les informations sur leurs types.

Le répertoire *templates* contient lui toutes les pages html de présentation, utilisées par les fichiers applicatifs présents à la racine.

Les fichiers d'initialisation et de configuration se trouvent, eux, dans le répertoire *inc*. On y trouve également toutes les classes et services mis à disposition pour Flash ainsi que toutes les classes externes utilisées.



4.8.5.2 DOCUMENTATION DU CODE

Toutes les classes métier ainsi que les classes « Remote » ont été documentées à l'aide des tags standards similaires à la JavaDoc. Ceci afin de pouvoir utiliser l'utilitaire PhpDocumentor qui permet de générer toute la documentation des classes nécessaire à la reprise du code existant par une personne externe. La documentation créée répertoriant toutes les informations nécessaires est disponible à la racine du CD annexé, dans le répertoire phpDoc. Un exemple de cette documentation générée est disponible en annexe.

```

/**
 *
 * @author Andre Lambelet
 * @package BusinessClasses
 * @since 01.07.2009
 *
 */
class Member {

    public $username;
    public $role;
    public $role_id;
    public $lastsession;
    public $facebookUid;
    public $globalHighscore;

    public static $MYSQL_TABLE = "member";
    public static $FB_PREFIX = "facebook_";

    public function Member ($username = "", $roleid = "", $lastsession = "", $facebookUid = "", $globalHighscore = "")
    {
        if (! empty($username))      $this->username      = $username;
        if (! empty($roleid))        $this->role_id       = $roleid;
        if (! empty($lastsession))   $this->lastsession    = $lastsession;
        if (! empty($facebookUid))   $this->facebookUid   = $facebookUid;
        if (! empty($globalHighscore)) $this->globalHighscore = $globalHighscore;
    }

    /**
     * Retourne un membre par son userid
     * @param String $userid
     * @return Member
     */
    public static function getSingleMember($username){

        $members = Member::getObjects(Member::show($username));
        if (count($members) >0)
            return $members[0];
        else
            return null;
    }
}

```

Source 12 - Exemple phpDoc, classe Member



5 GESTION DE PROJET

5.1 DÉROULEMENT

Ce travail de Bachelor a été réalisé en 361 heures réparties sur un total de 12 semaines. Durant cette période ont été menées toutes les phases d'un projet, du démarrage aux tests en passant par le développement et la rédaction des documents.

L'analyse du projet permettant la rédaction du cahier des charges et la planification des différentes phases suivantes a été effectuée durant les deux premières semaines. Parallèlement à ceci, des recherches ont été effectuées quand aux possibilités de couplage entre Flash et Facebook.

Deux semaines supplémentaires ont ensuite été accordées à une analyse plus approfondie du projet afin de définir les différents aspects de l'application. Le déroulement du jeu ainsi que le schéma de la base de données ont alors été définis. Des tests réels d'intégration de Flash et PHP à Facebook ont été menés afin de découvrir les solutions idoines.

Les six semaines développement et intégration ont ensuite débuté. Ponctuées par des échéances régulières, elles ont laissé place à une semaine de rédaction de la documentation.

En fonction de l'avancement du design Flash, la documentation a dû être effectuée en avance afin de ne pas être bloqué en fin de projet. Après avoir rédigé le premier jet de la documentation il a été possible de passer à l'amélioration du Jeu Flash créé par Hui Duan. En effet, le designer n'étant plus disponible en fin de projet, la correction des bugs et les finitions ont dû être effectuées par mes soins en fin de projet.



5.2 RÉPARTITION DES TÂCHES

Le projet E-creation 4 Drill on Facebook étant de plus grande envergure que la charge de travail prévue pour ce travail, une partie du développement a été prise en charge par un assistant de la HES-SO. Hui Duan s'est donc occupé du design et du développement Flash afin de permettre de mettre l'accent sur la gestion dynamique des types de questions. J'ai tout de même eu l'occasion d'apprendre cette nouvelle technologie en fin de projet, comme expliqué dans la section précédente.

5.3 PLANIFICATION

Selon la planification du projet E-creation 4, la fin du projet est prévue en octobre 2009. Dès lors que le nombre d'heures affectées à ce travail était de 360 du 11 mai au 03 août 2009, la planification a dû être rééchélonnée afin de correspondre à ces critères.

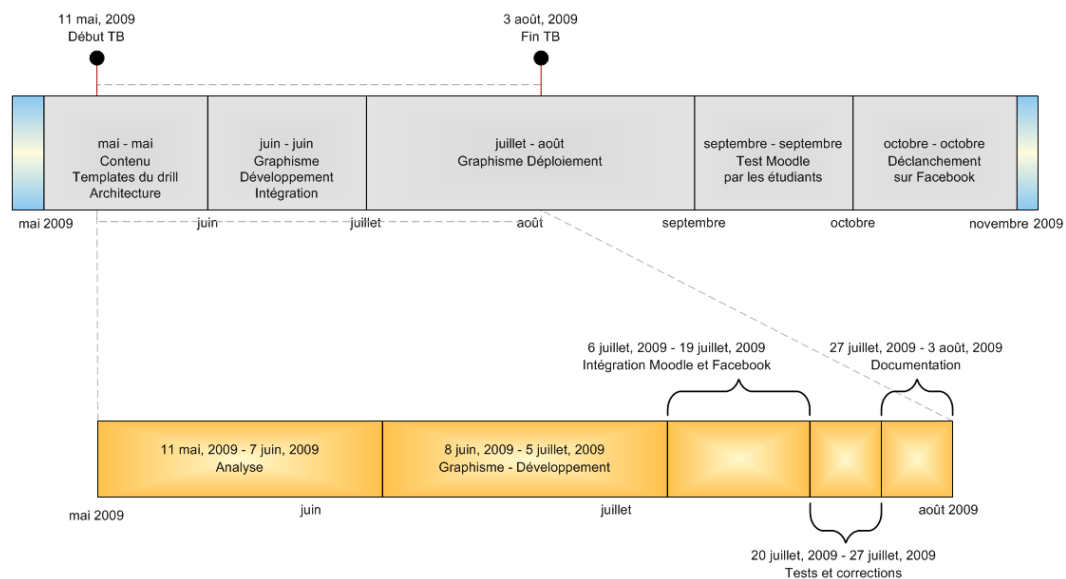


Schéma 15 – Planification initiale

Selon la planification initiale disponible en annexe et reproduite sur le schéma précédent, quelques adaptations ont dû être effectuées. Au niveau de la charge prévue au sein du travail de Bachelor aucun retard n'a été pris, comme vous pourrez le constater au chapitre « Bilan des heures effectuées ».



En fonction du développement Flash qui s'est effectué plus tardivement que prévu, les deux dernières phases ont été inversées afin de ne pas être bloqué en fin de projet. La documentation a donc été effectuée du 17 au 25 juillet afin de n'avoir plus que quelques détails à corriger une fois la phase "Tests et corrections" effectuée.

Les différentes échéances y relatives ont bien entendu été déplacées mais la durée des phases n'a subi aucun changement de grande envergure.

Une autre modification concernant la phase d'intégration à Moodle et Flash a été effectuée. Celle-ci s'est intégrée directement au développement car il était nécessaire de pouvoir intégrer dès le début les données de ces deux plateformes.

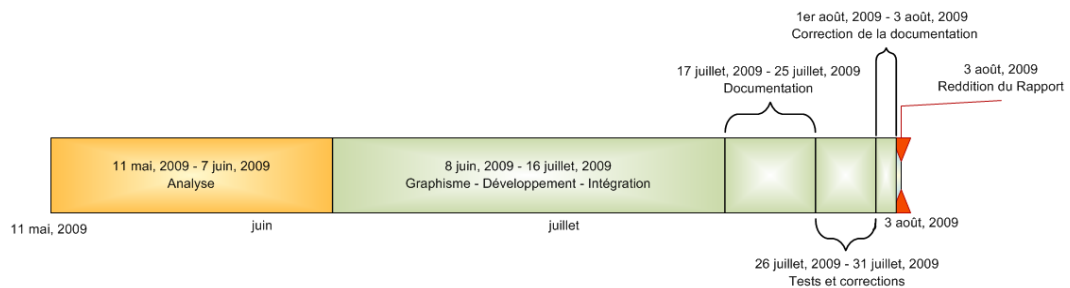


Schéma 16 - Planification finale

Nous pouvons donc constater que la planification initiale a été correctement effectuée. Une amélioration aurait pu être faite en ayant les dates exactes de disponibilité du designer Flash.

5.4 SUIVI HEBDOMADAIRE

Durant tout le projet, des séances hebdomadaires ont été planifiées chaque début de semaine afin de communiquer l'avancement du projet aux autres membres, valider les choix effectués durant la semaine, définir les points d'action futurs et faire le point sur la planification. Vous trouverez les PV de séance en annexe.



5.5 SAUVEGARDES

Des sauvegardes ont été effectuées à intervalles réguliers. Elles sont consultables sur le CD annexé.

5.6 BILAN DES HEURES EFFECTUÉES

Le temps de travail a été répertorié quotidiennement en fonction de la catégorie (analyse, développement et tests, documentation ou divers) afin de permettre une analyse poussée des heures effectuées.

Selon les chiffres que vous trouverez dans l'annexe « Heures effectuées », nous pouvons observer que, comme spécifié plus haut, les durées des différentes phases ont été correctement respectées. La partie développement ayant été terminée avec quelques jours d'avance il a été possible de commencer la documentation sur une plage plus longue.

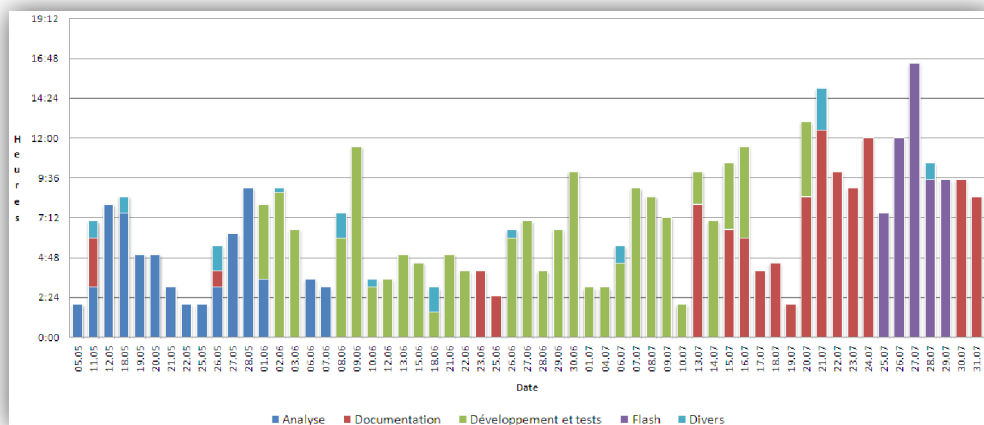


Tableau 4 - Graphique des heures / catégories par jour

Nous pouvons remarquer l'inversion qui a été effectuée au niveau des deux dernières phases, la partie Flash ayant été réalisée après la documentation. Vous noterez également les séances hebdomadaires répertoriées sous la catégorie "divers".



La répartition du temps passé pour chaque catégorie est cohérente. Par rapport à un projet standard, la part de l'analyse pourrait paraître quelque peu faible, mais étant donné que le projet E-creation 4 était déjà démarré, il restait uniquement l'analyse technique à effectuer.

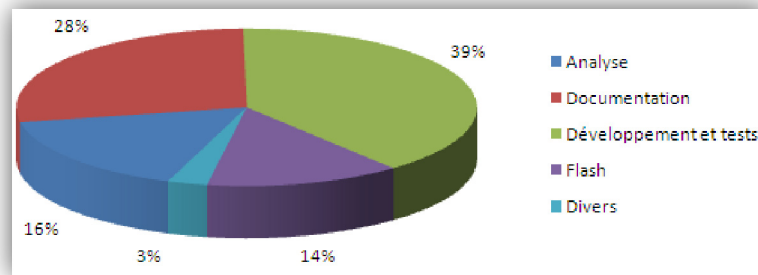


Tableau 5 - Graphique des proportions par catégorie

5.7 TRAVAIL RESTANT

Selon le cahier des charges disponible en annexe, tous les éléments requis ont été implémentés avec succès.

5.8 BILAN FINAL

Le projet ayant été terminé dans les temps et de manière exhaustive par rapport aux attentes définies dans le cahier des charges nous pouvons tirer un bilan très positif de ce projet. Les améliorations proposées n'étant pas vitales au niveau de l'application, celle-ci peut être considérée comme ayant dépassé le stade du prototype. Il a d'ailleurs été proposé de la mettre en production quelque peu en avance par rapport au planning initial du projet E-creation 4, soit dans le courant du mois d'août.



6 SATISFACTION PERSONNELLE

J'ai dès le départ été motivé par le thème de ce projet. Le fait de devoir travailler avec des plateformes de réseau social et d'apprentissage en ligne possédant chacune leurs spécificités ainsi que la nécessité d'apprendre une nouvelle technologie qu'est Flash, plus particulièrement le langage ActionScript 3.0, version avec laquelle je n'avais jamais eu l'occasion de travailler auparavant, a apporté un intérêt supplémentaire. Toujours à la recherche de nouveaux challenges, ces nouveautés m'ont permis d'œuvrer sur un sujet de travail de Bachelor adapté à mes attentes.

J'ai donc dû me familiariser, entre autres, à l'utilisation de l'API Facebook dont plusieurs syntaxes différentes et pas forcément aisément identifiables en découlent. Grâce à la documentation fournie j'ai pu acquérir de nouvelles connaissances que je compte utiliser dans un avenir proche.

La correction des bugs et amélioration du jeu Facebook m'a permis de m'immiscer dans l'environnement Flash encore inconnu auparavant. Après avoir passé quelques heures aux côtés d'Hui Duan afin de comprendre le fonctionnement de ce langage, j'ai pu me former convenablement afin de maîtriser ce nouveau langage.

Ce projet me permet désormais de bénéficier d'une expérience de plus dans la gestion d'un projet individuel relativement conséquent ainsi que d'ajouter de nouvelles compétences techniques à mon portfolio.

Rétrospectivement, ma satisfaction personnelle est grande. La liberté de mouvement m'ayant été accordée et la bonne entente au sein du team de projet original de E-creation 4 y ont grandement contribué. Que ce soit au niveau de la gestion du projet ou tout simplement de l'ambiance de travail sur le site du TechnoArk, il n'y a eu aucun accroc au bon déroulement de ce travail de Bachelor qui fût une agréable expérience.



7 DÉCLARATION SUR L'HONNEUR

Je déclare, par ce document, que j'ai effectué le travail de Bachelor ci-annexé seul, sans autre aide que celles dûment signalées dans les références, et que je n'ai utilisé que les sources expressément mentionnées. Je ne donnerai aucune copie de ce rapport à un tiers sans l'autorisation conjointe du RF et du professeur chargé du suivi du travail de Bachelor, y compris au partenaire de recherche appliquée avec lequel j'ai collaboré, à l'exception des personnes qui m'ont fourni les principales informations nécessaires à la rédaction de ce travail et que je cite ci-après :

- M. Florian Doche, professeur à la HES-SO Valais
- M. Bernard Crettenand, assistant à la HES-SO Valais
- M. Hui Duan, assistant à la HES-SO Valais

Sierre, le 03.08.2009

André Lambelet



8 BIBLIOGRAPHIE

8.1 GÉNÉRAL

- API : <http://en.wikipedia.org/wiki/Api>
- Smarty : <http://www.smarty.net/>
- MySQL : <http://www.mysql.com/>
- PHP : <http://www.php.net/>
- SOAP : <http://fr.wikipedia.org/wiki/SOAP>
- JavaDoc : <http://fr.wikipedia.org/wiki/Javadoc>
- Service Web : <http://www.alaide.com/dico.php?q=Web+Service&ix=3808>

8.2 FACEBOOK

- Adobe Developer Connection : <http://www.adobe.com/devnet/facebook/>
- Choosing between an FBML or IFrame Application :
http://wiki.developers.facebook.com/index.php/Choosing_between_an_FBML_or_IFrame_Application
- An Introduction to FBML : <http://20bits.com/articles/an-introduction-to-fbml/>
- Comparing Flash iFrame and FBML Facebook applications :
http://www.adobe.com/devnet/facebook/articles/iframe_fbml_flash_platform_comparison.html
- Facebook Developers : <http://developers.facebook.com>

8.3 FLASH

- Action Script 2.0 Tutorials : <http://nidoweb.com/as3.php>
- Loading data into Flash : <http://www.flash-db.com/Tutorials/loading/>
- Populate Combo Box : <http://www.flash-db.com/Board/index.php?topic=18599.0>
- Populate Datagrid : <http://www.flash-db.com/Board/index.php?topic=18601.0>
- Facebook Bridge : <http://components.zerofractal.com/FacebookBridge/index.html>
- FB App Canvas Pages, I think I'd Use IFrames : <http://www.ccheever.com/blog/?p=10>



8.4 JAVASCRIPT

- Htmlentities : <http://phpjs.org/functions/htmlentities:425>
- Htmlspecialchars : <http://phpjs.org/functions/htmlspecialchars:426>
- Regular expression validation : <http://javascript.internet.com/forms/regexp-validation.html>

8.5 PHP

- Amfphp : <http://www.amfphp.org/docs/helperclasses.html>
- Fonction currentTimeMillis() : <http://benjisimon.blogspot.com/2007/09/currenttimemillis-in-php.html>
- ConvertLatin1ToHtml : <http://files.codes-sources.com/fichier.aspx?id=47552&f=team.php>
- Classe Upload : <http://www.verot.net/>
- XSL de PHP4 à PHP5 : <http://alexandre.alapetite.fr/doc-alex/xslt-php4-php5/>

8.6 XML

- Transformations XSLT : <http://classes.scriptsphp.org/article.Transformations-XSLT-en-PHP-5>
- Génération de XML à partir d'HTML : <http://www.seusoft.com/Html2Xml01.html>
- XmlForm : <http://www.phpclasses.org/browse/package/468.html>

8.7 IMAGES

- EasyPhp : http://www.customxp.net/PngFactory/png/_thumb/7906-durentoaine-easyPHP.png
- Apache : http://www.alunys.com/page/52/@fck_resource@/apache_logo.jpg
- JavaScript : http://www.setupgroup.com/images/javascript_icon.gif
- PHP classes : http://files.phpclasses.org/graphics/elephant_logo.gif
- PHP : <http://fr.wikipedia.org/wiki/PHP>
- PhpMyAdmin : http://www.phpmyadmin.net/home_page/images/logo.png
- XML : http://www.informanews.net/tests/fluxRSS/xml_logo.gif
- XML : http://library.case.edu/digitalcase/images/doc_xml_icon.png
- XSLT : http://ditchnet.org/aquaxsl/images/aquaxsl_icon.gif



9 ANNEXES

9.1 CAHIER DES CHARGES

Cahier des charges

Projet : HES-SO Driller

HES-SO Valais // Wallis 2009 - Travail de Bachelor
Lambelet André

INTRODUCTION

Dans le cadre du projet e-cr  ation 4, Mme Glassey souhaite d  velopper pour les   tudiants de la HES-SO un drill de programmation disponible sur la plateforme Cyberlearn¹ (*Moodle*) ainsi que sur le site communautaire *facebook*².

Ce projet (HES-SO Driller) a pour but de motiver les   tudiants lors de l'apprentissage de la programmation de base par le c  t   ludique du jeu mais   galement le c  t   comp  titif et communautaire de *facebook*. Le contexte innovant ainsi que la collaboration avec deux plateformes d  j   connues des   tudiants devrait permettre de favoriser l'utilisation de ce drill afin de pousser le public cible    la pratique r  guli  re des exercices propos  s.

TABLE DES MATI  RES

Introduction	1
Description g��n��rale.....	2
Choix de technologie.....	2
Le jeu.....	2
Administration du jeu	2
Description du projet	2
Utilisateurs cible.....	2
Les ��tudiants	2
Les professeurs	3
Facebook.....	3
Sp��cificit��s	3
Structure des cours	3
Structure d'un cours (quizz)	4
Contenu du m��ta-cours	4
Contenu d'un cours	4
Couplage avec le LMS <i>Moodle</i>	4
Authentification	4
Proposition de questions	5
Cr��ation d'un cours	5
Gestion des scores	5
Publication des r��sultats	5
Evolution de projet.....	5
Planning	5
Signatures	5

¹ <http://cyberlearn.hes-so.ch>

² <http://www.facebook.com>



DESCRIPTION GÉNÉRALE

CHOIX DE TECHNOLOGIE

HES-SO Driller sera composé de deux parties distinctes:

- Le Jeu
- Administration du jeu

LE JEU

Afin de garantir un jeu dynamique et motivant, cette partie du projet sera principalement développée avec Adobe Flash. L'intégration du jeu Flash avec les différentes plateformes se fera par des pages PHP.

ADMINISTRATION DU JEU

La plateforme d'administration sera une interface web développée en PHP.

DESCRIPTION DU PROJET

HES-SO Driller devra offrir les points suivants:

- Possibilité aux professeurs de créer et personnaliser un cours
- Bloquer l'accès à un cours par un mot de passe
- Quatre types de quiz prédéfinis à choix lors de la création
- Gestion des scores et comparaisons aux autres utilisateurs
- Statistiques des résultats obtenus par tous les étudiants
- Possibilité aux étudiants de proposer ou commenter une question

UTILISATEURS CIBLE

Deux types d'utilisateurs sont visés :

1. Les étudiants touchés par la programmation
2. Les professeurs de programmation
3. Les membres de la communauté *facebook*

LES ÉTUDIANTS

Les buts du projet pour un étudiant sont :

- Tester ses connaissances
- Bénéficier d'un support attrayant
- Avancer à son rythme

L'accès à HES-SO Driller depuis la plateforme *Moodle* se fera uniquement par un wrapper contenant le jeu. L'étudiant devra ensuite choisir le cours désiré.



LES PROFESSEURS

Les buts du projet pour un professeur sont:

- Améliorer la participation
- Apport de nouvelles questions (outil évolutif)

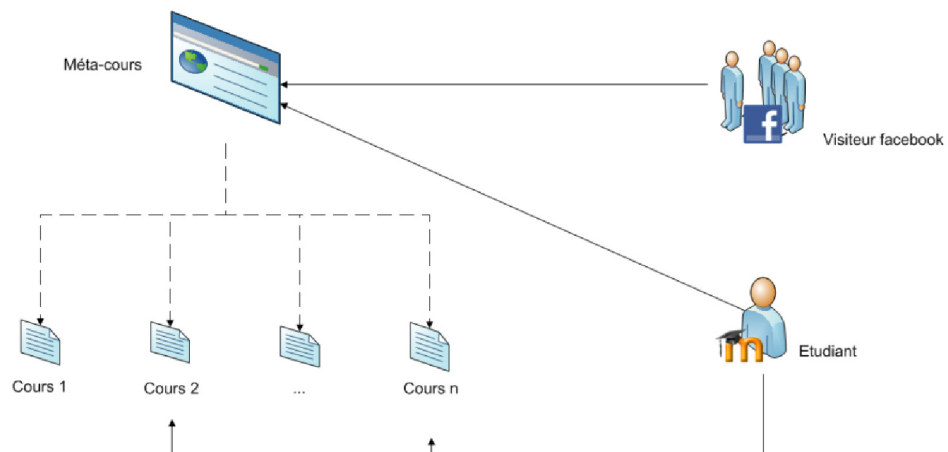
Les professeurs bénéficieront d'un accès spécifique à la console d'administration afin de gérer les cours, jeux et questions ainsi que de consulter les statistiques des résultats des étudiants.

FACEBOOK

L'accès à l'application se fera par une application *facebook* qui proposera à tous les membres de cette plateforme de participer au méta-cours (cf. ch. Structure des cours). Ils n'auront pas la possibilité de choisir un cours en particulier.

SPÉCIFICITÉS

STRUCTURE DES COURS



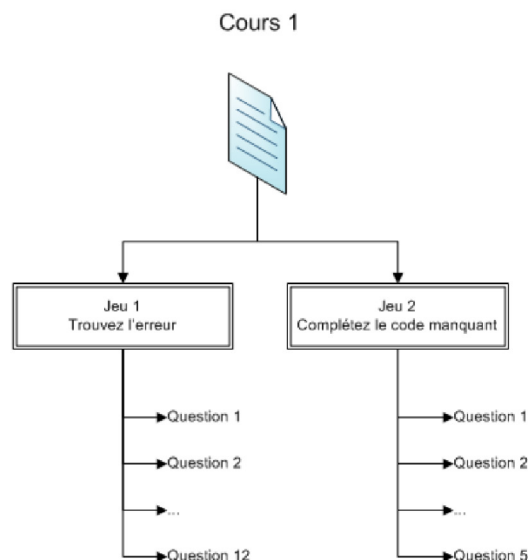
Un méta-cours sera composé de toutes les questions que les professeurs auront ajoutées. Ce méta-cours sera proposé à tous les élèves ainsi qu'aux visiteurs de *facebook*.

Les professeurs auront alors la possibilité de créer un cours personnalisé. Ce cours sera protégé ou non par un mot de passe.



STRUCTURE D'UN COURS (QUIZZ)

Un cours sera composé de plusieurs jeux différents à choisir parmi les types proposés, selon le schéma ci-dessous représentant la structure du cours intitulé "Cours 1" et composé de 2 jeux différents.



CONTENU DU MÉTA-COURS

Le méta cours sera constitué de tous les types de jeux, chacun comprenant un nombre de questions (définissable par l'administrateur de l'application). Les questions seront choisies au hasard parmi toutes les questions disponibles pour le niveau choisi par le visiteur.

CONTENU D'UN COURS

Les cours créés par les professeurs seront quand à eux totalement définis par leur auteur. Ils devront effectivement choisir quel type de jeux ils désirent proposer ainsi que chaque question et l'ordre dans lequel elles seront posées (tirage aléatoire également possible). Il n'y aura pas de choix du niveau possible.

COUPLAGE AVEC LE LMS MOODLE

L'accès au jeu par *Moodle* se fera par un wrapper. Il ne sera malheureusement pas possible de coupler l'authentification *Moodle* avec le jeu. L'étudiant aura donc la possibilité d'accéder au méta-cours, aux cours non-protégés ainsi qu'à ceux dont il connaît le mot de passe.

AUTHENTIFICATION

L'authentification au jeu sera proposée aux visiteurs afin de leur permettre de visualiser leurs anciens cours, leur score et le classement par rapport aux autres élèves. De plus, cela permettra à l'étudiant de ne pas avoir à entrer le mot de passe à chaque fois qu'il voudra passer un même cours.



PROPOSITION DE QUESTIONS

Les étudiants auront la possibilité de proposer une question aux professeurs. Pour ce faire ils devront créer leur questions et ensuite choisir à quel professeurs ils aimeraient la proposer. Un choix indéterminé permettra à n'importe quel professeur de la valider et donc d'en devenir responsable.

CRÉATION D'UN COURS

Lors de la création d'un cours, chaque professeur aura la possibilité de choisir parmi les types de jeux disponibles. Ils devront ensuite choisir les questions désirées parmi le pool de questions général. Ils pourront ajouter une question au méta-cours ou directement à leur cours. Dans ce cas la question sera automatiquement ajoutée au méta-cours. Cette procédure permettra d'avoir un méta-cours qui soit toujours à jour par rapport aux cours afin de proposer un drill public évoluant au fil du temps.

GESTION DES SCORES

Au vu de l'évolution des questions, les scores devant être comparables, une gestion du niveau (1-10) de chaque question sera mise en place. Ceci permettra de proposer trois niveaux aux participants du méta-cours :

- Débutant ★
- Moyen ★ ★
- Avancé ★ ★ ★

De cette manière, la comparaison des scores sera toujours équivalente, peu importe les questions posées dans le cours.

PUBLICATION DES RÉSULTATS

Après chaque passage du méta-cours l'étudiant aura la possibilité de publier ses résultats sur *facebook*. L'étudiant sera par contre automatiquement visible sur le classement général mis à disposition des autres étudiants ayant un accès par la plateforme *Moodle*.

EVOLUTION DE PROJET

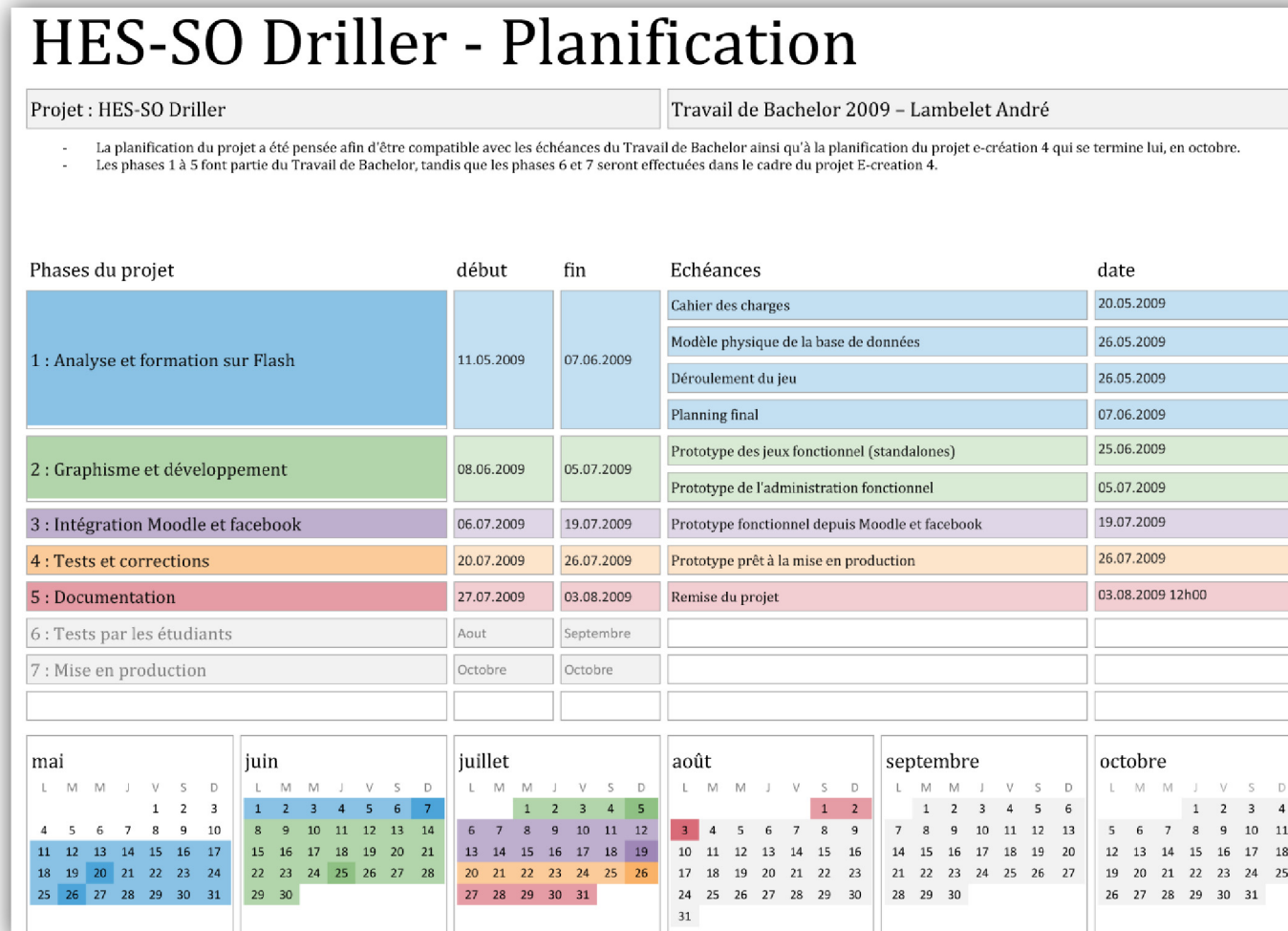
Ce projet est premièrement axé sur l'objectif de proposer des types de jeux de programmation. La plateforme devra cependant être pensée afin de pouvoir évoluer afin de proposer des cours sur des sujets différents.

PLANNING

Le planning du projet se trouve en annexe (document "HES-SO Driller – Planification").



9.2 PLANIFICATION INITIALE



9.3 PROCÈS-VERBAUX

HESSODriller minutes

Séance d'introduction			
05.05.2009		10h00	TechnoArk
Organisée par	Mme Nicole Glassey		
Type de réunion	Hebdomadaire		
Participants	Mme Nicole Glassey, M. Florian Doche, M. André Lambelet		
Définition du Travail de Bachelor			
	Mme Nicole Glassey		
Discussion	Présentation du Projet "Drill on Facebook". Discussion des possibilités de drills.		
Conclusions	Quatre types de quizz seront mis à disposition Les élèves pourront proposer des questions qui devront être validées par les professeurs		
Points d'action		Personne responsable	Délai
Organiser une réunion avec B. Crettenand et H. Duan		F. Doche	a.s.a.p.



HESSODriller minutes

Séance			
05.05.2009		14h00	TechnoArk
Organisée par	F. Doche		
Type de réunion	Extraordinaire		
Participants	N. Glassey, F. Doche, B. Crettenand, H. Duan, A. Lambelet		
Etude des possibilités d'interaction avec Moodle			
	B. Crettenand		
Discussion	Possibilité d'intégration à Moodle		
Conclusions	Il est possible d'afficher le jeu dans un cadre au sein de facebook. Par contre il n'est pas possible d'avoir un Single Login entre Moodle et le jeu flash.		
Etude des possibilités au niveau de Flash			
	H. Duan		
Discussion	Il n'y a pas de restrictions au niveau du design de Flash		
Conclusions	Prévoir le planning en fonction des disponibilités de Huan. Le jeu sera créé avec un design basic par André et ensuite Huan s'occupera de l'améliorer.		
Points d'action		Personne responsable	Délai
Se documenter sur les possibilités d'interactions Flash-Facebook		A. Lambelet	a.s.a.p.



HESSODriller minutes

Séance			
11.05.2009		10h00	TechnoArk
Organisée par	N. Glassey		
Type de réunion	Hebdomadaire		
Participants	N. Glassey, A. Lambelet		
Etude des possibilités d'interaction avec Moodle			
	N. Glassey		
Discussion	<ol style="list-style-type: none">1. Comment gérer les scores alors que les questions changent?2. Présentation de la structure des cours		
Conclusions	<ol style="list-style-type: none">1. 3 niveaux seront définis afin de pouvoir classer les questions et permettre une comparaison entre les scores. Le fait que des questions seront ajoutées au fur et à mesure pourra également motiver les étudiants et utilisateurs à passer les quizz plus souvent.2. Un "super-cours" sera formé de toutes les questions. Un professeur pourra créer un cours et y reprendre les questions voulues. Par contre il ne pourra ajouter une question qu'au "super-cours". Cela permettra d'avoir un pool de questions disponibles pour le quizz public et pour les autres professeurs.		
Points d'action		Personne responsable	Délai
Rédaction du Cahier des charges		A. Lambelet	20.05.2009
Etablissement du planning		A. Lambelet	20.05.2009



HESSODriller minutes

Séance			
18.05.2009		16h00	TechnoArk
Organisée par	N. Glassey		
Type de réunion	Hebdomadaire		
Participants	N. Glassey, A. Lambelet		
Feedback de la semaine			
Feedback	<ul style="list-style-type: none">- Rédaction d'une analyse concernant l'architecture à utiliser.- Configuration d'un serveur en DMZ (à terminer)		
Points d'action en cours		Etat	Délai
Rédaction du Cahier des charges		En cours	20.05.2009
Etablissement du planning		En cours	20.05.2009
Etat du projet			
	N. Glassey		
Discussion	<ol style="list-style-type: none">1. Deux modèles de cours sont déjà définis. Le premier ou l'utilisateur doit cliquer sur la faute dans le code et le deuxième ou l'élève doit compléter un code en cliquant sur la séquence juste parmi les choix.2. Les statistiques devront être disponibles pour les professeurs3. Une possibilité devra être donnée aux utilisateurs de commenter les questions4. Planification pas nécessairement avec MS Project.		
Conclusions	<ol style="list-style-type: none">1. -2. Toutes les données des quizz (réponses) seront gardées afin de permettre aux professeurs d'avoir un feedback sur la pertinence de leurs questions.3. Une interface devra être prévue comme amélioration (pas prioritaire).4. Prévision des deadlines au minimum toutes les 3 semaines.		
Points d'action		Personne responsable	Délai
Création du modèle de la base de données		A. Lambelet	26.05.2009
Rédaction du déroulement du jeu		A. Lambelet	26.05.2009
Prochaine séance		Mardi 26.05.2009	Matin



HESSODriller minutes

Séance			
26.05.2009		15h00	TechnoArk
Organisée par	N. Glassey		
Type de réunion	Hebdomadaire		
Participants	N. Glassey, H. Duan, A. Lambelet		
Feedback de la semaine			
Feedback	<ul style="list-style-type: none">• Architecture entre Flash et Facebook validée• Planning global validé		
Points d'action en cours		Etat	Délai
Rédaction du Cahier des charges		Terminé	-
Modèle de la base de données		Terminé	-
Etablissement du planning		Validé	-
Architecture entre Flash et Facebook		Validée	-
Etat du projet			
Informations	<ul style="list-style-type: none">• Relecture et correction du cahier des charges• Présentation de la navigation possible pour un étudiant• Présentation du schéma d'un cours• Présentation du schéma physique de la Base de Données		
Décisions	<ul style="list-style-type: none">• La visibilité dans les classements est obligatoire sur Moodle ainsi que pour les utilisateurs facebook mais pas la publication du résultat sur facebook depuis Moodle• Une limite de temps devra être définie par niveaux• Un seul classement général sera généré (niveaux mélangés)• Trois types de jeux définis:<ol style="list-style-type: none">1. Réponse à une question2. Complétion du code manquant3. Recherche d'erreur		
Points d'action		Personne responsable	Délai
Corriger le modèle de la base de données		A. Lambelet	02.06.2009
Rédaction du déroulement du jeu		A. Lambelet	26.05.2009
Correction du Cahier des charges		A. Lambelet	02.06.2009
Prochaine séance		Mardi 02.06.2009	Journée



HESSODriller minutes

Séance			
02.06.2009		15h00	TechnoArk
Organisée par	N. Glassey		
Type de réunion	Hebdomadaire		
Participants	N. Glassey, H. Duan, A. Lambelet		
Feedback de la semaine			
Feedback	<ul style="list-style-type: none">• Mise en place de la structure du serveur web• Tests d'interaction Flash-Facebook sans résultats• Début de la création des classes		
Points d'action en cours		Etat	Délai
Corriger le modèle de la base de données		Terminé	-
Rédaction du déroulement du jeu		Terminé	-
Correction du Cahier des charges		Terminé	-
Etat du projet			
Informations	<ul style="list-style-type: none">• Reste à définir la méthode de stockage du contenu des questions et réponses		
Décisions	<ul style="list-style-type: none">• Division en deux parties distinctes : Code (André) – Design (Hui)		
Points d'action		Personne responsable	Délai
Préparer quelque chose à présenter		A. Lambelet	08.06.2009
Formation sur Flash AS3		A. Lambelet	a.s.a.p.
Prochaine séance		Lundi 08.06.2009	Après-midi



HES-SO Driller minutes

Séance			
08.06.2009		14h00	TechnoArk
Organisée par	N. Glassey		
Type de réunion	Hebdomadaire		
Participants	N. Glassey, F. Doche, H. Duan, A. Lambelet		
Feedback de la semaine			
Feedback	<ul style="list-style-type: none">• Bon avancement dans les différents points de la structure<ul style="list-style-type: none">- Administration- liaison Flash-Facebook- liaison Flash-Base de données		
Points d'action en cours		Etat	Délai
-		-	-
Etat du projet			
Informations	<ul style="list-style-type: none">• Récupération des données Facebook par Flash => ok• Web Service de vérification de l'authentification => ok, à tester sur le serveur de moodle• Interface d'administration en cours		
Décisions	<ul style="list-style-type: none">• Protos-écrans à définir pour chaque étape du jeu• Définition du schéma XML du premier jeu• Le logo de l'application sera créé par Hui• La partie développement des jeux sera prise en charge par Hui afin de créer une structure plus avancée.		
Points d'action		Personne responsable	Délai
Obtenir les accès sur la base moodle		A. Lambelet	a.s.a.p.
Protos-écrans		A. Lambelet	15.06.2009
Schéma XML du premier jeu		A. Lambelet	15.06.2009
Prochaine séance		Lundi 22.06.2009	Journée
Prochaine séance avec F. Doche		Mardi 23.06.2009	Après-midi



HES-SO Driller minutes

Séance		
23.06.2009	14h00	TechnoArk
Organisée par	N. Glassey	
Type de réunion	Hebdomadaire	
Participants	N. Glassey, F. Doche, H. Duan, A. Lambelet	
Feedback de la semaine		
Feedback	<ul style="list-style-type: none">Mise en place des types de questions dynamiques par XML, XSLT et XSDDéfinition des signatures des methodes nécessaires pour le webserviceDémonstration du login Flash lié à moodle	
Points d'action en cours	Etat	Délai
Obtenir les accès sur la base moodle	Terminé	-
Protos-écrans	Terminé	-
Schéma XML du premier jeu	Terminé	-
Etat du projet		
Informations	<ul style="list-style-type: none">La plupart des éléments de l'application sont définisLes proto-écrans ont été définis avec HuiLe design principal ainsi que les icônes sont déjà créés par Hui	
Décisions	<ul style="list-style-type: none">Le nombre de réponses devra pouvoir être différent pour chaque questionAfin de garder une mise en page des questions, tinyMCE sera intégré aux champs textes lors de la modification des questionsAfin de prévoir la charge des utilisateurs, un document sur la configuration recommandée devra être établiAu vu des heures de travail restantes, Un deuxième jeu pourra être intégrer si le temps le permet.	
Points d'action	Personne responsable	Délai
Obtenir un wrapper sur moodle	A. Lambelet	a.s.a.p.
Document sur la config minimale requise	A. Lambelet	a.s.a.p.
Prototype fonctionnel de la partie Facebook	H. Duan	05.07.2009
Prochaine séance		
	Lundi 06.07.2009	13h30



HES-SO Driller minutes

Séance			
06.07.2009		15h00	TechnoArk
Organisée par	F. Doche		
Type de réunion	Hebdomadaire		
Participants	F. Doche, H. Duan, A. Lambelet		
Feedback de la semaine			
Feedback	<ul style="list-style-type: none">• Ajout de l'interface de gestion des types de questions• Des paramètres peuvent être passés depuis moodle afin d'authentifier l'utilisateur sans qu'il ait à s'authentifier une deuxième fois.		
Points d'action en cours		Etat	Délai
Obtenir un wrapper moodle		En cours	-
Document sur la configuration minimale requise		En cours	-
Prototype fonctionnel de la partie Facebook		En cours	05.07.2009
Etat du projet			
Informations	<ul style="list-style-type: none">• La partie Facebook n'est pas encore terminée mais le sera pour la fin de la semaine.• Problème rencontré lors de l'intégration de tinyMCE ainsi que pour la modification des questions		
Décisions	<ul style="list-style-type: none">• Pour les 2 prochaines semaines:<ul style="list-style-type: none">- Interactions facebook (inviter des amis, newsfeeds, etc.)- Correction de bugs- Documentation		
Points d'action		Personne responsable	Délai
Création d'un compte ftp pour Hui sur le serveur		A. Lambelet	a.s.a.p.
Prototype fonctionnel de la partie Facebook		H. Duan	05.07.2009
Prochaine séance		Lundi 20.07.2009	Après-midi



HES-SO Driller minutes

Séance			
21.07.2009		10h00	TechnoArk
Organisée par	N. Glassey		
Type de réunion	Hebdomadaire		
Participants	N. Glassey, H. Duan, A. Lambelet		
Feedback de la semaine			
Feedback	<ul style="list-style-type: none">• Présentation du jeu flash pour Moodle• Présentation de la partie administration terminée.• Présentation de l'état actuel du rapport final		
Points d'action en cours		Etat	Délai
Obtenir un wrapper Moodle		Terminé	-
Document sur la configuration minimale requise		En cours	03.08.2009
Prototype fonctionnel de la partie Facebook		OK pour Moodle	05.07.2009
Création d'un compte ftp pour Hui sur le serveur		Terminé	-
Etat du projet			
Informations	<ul style="list-style-type: none">• La partie Facebook n'est pas encore terminée, pour des raisons de rapidité, l'accent a été mis sur la partie Moodle qui nécessite plus de développement.• Discussions autour du Rapport final• Hui sera en vacances dès le 23.07.2009 jusqu'à la fin du travail. Les sources flash seront à disposition si nécessaire.		
Décisions	<p>Administration:</p> <ul style="list-style-type: none">- 10 Questions aléatoires par type de jeu pour le quizz général- Prochain type de jeu à développer : Trouvez l'erreur. L'ajout du code se fera ligne par ligne.- Finalement une seule limite de temps n'est nécessaire par type de question- Si le temps le permet, automatiser les droits d'accès à l'administration en fonction du profil Moodle <p>Flash:</p> <ul style="list-style-type: none">- Bouton "suggestion" à déplacer <p>Documentation :</p> <ul style="list-style-type: none">- Ajouter un manuel d'installation et de configuration du serveur et des services.		
Points d'action		Personne responsable	Délai
Terminer le premier jet du rapport final		A. Lambelet	27.07.2009
Prochaine séance		Début sem. prochaine	A définir



HES-SO Driller minutes

Séance			
28.07.2009		11h00	TechnoArk
Organisée par	N. Glassey		
Type de réunion	Hebdomadaire		
Participants	N. Glassey, A. Lambelet		
Feedback de la semaine			
Feedback	<ul style="list-style-type: none">• Ajout du nouveau type de jeu OK• Modifications du jeu Flash effectuées, quelques détails à régler et c'est terminé• Trois chapitres du rapport à terminer		
Points d'action en cours		Etat	Délai
Terminer le premier jet du rapport final		Terminé	-
Document sur la configuration minimale requise		Terminé	-
Etat du projet			
Informations	<ul style="list-style-type: none">• Problème d'authentification avec IE à régler• Dernies bugs dans le Flash à régler		
Décisions	<ul style="list-style-type: none">• Les 3 limites de temps par type de questions sont à laisser tel quels• Les réponses ne doivent pas apparaître dans un ordre aléatoire• Ajouter la consigne pour le type "trouvez l'erreur"		
Points d'action		Personne responsable	Délai
Reddition du rapport final		A. Lambelet	03.08.2009
Prochaine séance (présentation du projet)		28.08.2009	10h



9.4 HEURES EFFECTUÉES

Sem	date	Catégorie	description	Début	Fin	Pauses	Total	360:00
1	05.05.2009	Analyse	Séance d'introduction	11.00 h	12.00 h		1:00	359:00 h
1	05.05.2009	Analyse	Séance avec Bernard Crettenand et Roui	14.30 h	15.30 h		1:00	358:00 h
1	11.05.2009	Divers	Introduction au TB	09.00 h	10.00 h		1:00	357:00 h
1	11.05.2009	Analyse	Recherches sur le couplage Flash-Facebook	10.00 h	14.00 h	01.00 h	3:00	355:00 h
1	11.05.2009	Documentation	Mise en place des documents utiles	14.00 h	17.00 h		3:00	352:00 h
1	12.05.2009	Analyse	Choix de l'architecture (FBML vs iframe)	09.00 h	18.00 h	01.00 h	8:00	345:00 h
2	18.05.2009	Analyse	Choix de l'architecture et début du cahier des charges	08.00 h	16.00 h	00.30 h	7:30	338:00 h
2	18.05.2009	Divers	Séance hebdomadaire	16.00 h	17.00 h		1:00	337:00 h
2	19.05.2009	Analyse	Cahier des charges	10.00 h	15.00 h		5:00	332:00 h
2	20.05.2009	Analyse	Cahier des charges et planning	08.00 h	14.30 h	01.30 h	5:00	328:30 h
2	21.05.2009	Analyse	Modèle Physique de Données	17.00 h	20.00 h		3:00	325:30 h
2	22.05.2009	Analyse	User Process	13.00 h	15.00 h		2:00	323:30 h
3	25.05.2009	Analyse	Game Process	14.00 h	16.00 h		2:00	321:30 h
3	26.05.2009	Analyse	Préparation des documents	13.00 h	16.00 h		3:00	318:30 h
3	26.05.2009	Divers	Séance hebdomadaire	16.00 h	17.30 h		1:30	317:00 h
3	26.05.2009	Documentation	Documentation	17.30 h	18.30 h		1:00	316:00 h
3	27.05.2009	Analyse	Correction des documents	07.45 h	14.00 h		6:15	309:45 h
3	28.05.2009	Analyse	Définition de l'architecture	08.00 h	17.00 h		9:00	300:45 h
4	01.06.2009	Développement	Mise en place du serveur	12.00 h	15.00 h		3:00	297:45 h
4	01.06.2009	Développement	Création de l'application facebook	15.00 h	16.30 h		1:30	296:15 h
4	01.06.2009	Analyse	Tests d'intégration facebook	20.00 h	23.30 h		3:30	292:45 h
4	02.06.2009	Développement	Mise en place de l'architecture PHP	07.30 h	12.00 h		4:30	288:15 h
4	02.06.2009	Développement	Création des classes avec méthodes standard	13.00 h	17.30 h	00.15 h	4:15	284:15 h
4	02.06.2009	Divers	Séance Hebdomadaire	15.00 h	15.15 h		0:15	284:00 h
4	03.06.2009	Développement	Interfaces graphiques (gestion des questions)	08.00 h	14.30 h		6:30	277:30 h
4	06.06.2009	Analyse	Tests d'intégration facebook	18.30 h	22.00 h		3:30	274:00 h
4	07.06.2009	Analyse	Tests d'intégration facebook	12.30 h	15.30 h		3:00	271:00 h
5	08.06.2009	Développement	Création du Webservice d'authentification	09.00 h	11.30 h		2:30	268:30 h
5	08.06.2009	Divers	Séance hebdomadaire	14.00 h	15.00 h		1:00	267:30 h
5	08.06.2009	Divers	Rédaction du PV	13.30 h	14.00 h		0:30	267:00 h
5	08.06.2009	Développement	Interface administration (gestion des questions)	14.00 h	19.00 h	01.30 h	3:30	265:00 h
5	09.06.2009	Développement	Interface administration	09.00 h	18.00 h	01.30 h	7:30	259:00 h
5	09.06.2009	Développement	Interface administration	22.00 h	02.00 h		4:00	255:00 h
5	10.06.2009	Divers	Séance avec Hui	10.30 h	11.00 h		0:30	254:30 h
5	10.06.2009	Développement	Création du schéma xsd	13.00 h	16.00 h		3:00	251:30 h
5	12.06.2009	Développement	tests d'intégration moodle + Schémas xsd	09.00 h	14.00 h	01.30 h	3:30	249:30 h
5	13.06.2009	Développement	Mise en place des webservices pour Flash	10.00 h	15.00 h		5:00	244:30 h
6	15.06.2009	Développement	Création d'XML depuis un formulaire html	08.00 h	12.30 h		4:30	240:00 h
6	18.06.2009	Divers	Revue de la planification	16.00 h	17.30 h		1:30	238:30 h
6	18.06.2009	Développement	Création de styles de boutons et implémentation	18.00 h	19.30 h		1:30	237:00 h
6	21.06.2009	Développement	formulaire dynamique (XML+XSLT)	17.00 h	22.00 h		5:00	232:00 h
7	22.06.2009	Développement	formulaire dynamique (XML+XSLT)	08.00 h	12.00 h		4:00	228:00 h
7	23.06.2009	Documentation	Documentation	08.00 h	12.00 h		4:00	224:00 h
7	25.06.2009	Documentation	Rédaction d'ébauches de chapitres	23.00 h	01.30 h		2:30	221:30 h
7	26.06.2009	Développement	Création des méthodes du webservice	07.30 h	11.00 h		3:30	218:00 h
7	26.06.2009	Divers	Rédaction du PV	11.00 h	11.30 h		0:30	217:30 h
7	26.06.2009	Développement	Synchronisation des comptes avec facebook	11.30 h	14.00 h		2:30	215:00 h
7	27.06.2009	Développement	Mise en place de formulaires au lieu de liens + génération Form2XML	12.00 h	15.00 h		3:00	212:00 h
7	27.06.2009	Développement	Interface gestion des types de questions	16.00 h	20.00 h		4:00	208:00 h
7	28.06.2009	Développement	Intégration de tinyMCE (ne fonctionne pas)	16.00 h	20.00 h		4:00	204:00 h



Sem	date	Catégorie	description	Début	Fin	Pauses	Total	360:00
8	29.06.2009	Développement	Création des webServices Flash	10.00 h	13.00 h		3:00	201:00 h
8	29.06.2009	Développement	Gestion des suggestions + Intégration de tinyMCE (pour autre que insertion de question)	13.30 h	17.30 h	00.30 h	3:30	198:00 h
8	30.06.2009	Développement	Synchro facebook, Confirmation suppressions, WebService pour les Ranking	07.30 h	19.00 h	01.30 h	10:00	189:30 h
8	01.07.2009	Développement	Optimisation du code (réduction des requêtes)	07.00 h	10.00 h		3:00	186:30 h
8	04.07.2009	Développement	Gestion des types de question	10.00 h	13.00 h		3:00	183:30 h
9	06.07.2009	Développement	Gestion des types de questions	13.30 h	19.00 h	01.00 h	4:30	180:00 h
9	06.07.2009	Divers	Séance avec Florian	15.00 h	16.00 h		1:00	179:00 h
9	07.07.2009	Développement	Edition des questions + Intégration TinyMCE pour les questions	07.30 h	18.00 h	01.30 h	9:00	171:30 h
9	08.07.2009	Développement	Modification des webservices + création cours moodle + authentification automatique	08.00 h	18.00 h	01.30 h	8:30	164:30 h
9	09.07.2009	Développement	Webservices	09.00 h	17.30 h	01.15 h	7:15	158:30 h
9	10.07.2009	Développement	WebServices	10.00 h	12.00 h		2:00	156:30 h
10	13.07.2009	Documentation	Structure et Intégration des docs existants	08.00 h	17.00 h	01.00 h	8:00	149:30 h
10	13.07.2009	Développement	Correction webservice + Documentation du code	17.30 h	19.30 h		2:00	147:30 h
10	14.07.2009	Développement	Séance avec Hui + Services web	14.00 h	21.00 h		7:00	140:30 h
10	15.07.2009	Documentation	Documentation du code (phpDocumentor)	07.00 h	15.00 h	01.30 h	6:30	135:30 h
10	15.07.2009	Développement	Méthode d'authentification à un cours + gestion des profs et admins	15.00 h	19.00 h		4:00	131:30 h
10	16.07.2009	Documentation	Redéfinition de la structure + ébauche de chapitres	08.00 h	12.00 h		4:00	127:30 h
10	16.07.2009	Développement	Dev avec Hui, Web services	13.30 h	19.00 h		5:30	122:00 h
10	16.07.2009	Documentation	Documentation du code	20.00 h	22.00 h		2:00	120:00 h
10	17.07.2009	Documentation	Documentation	10.00 h	14.00 h		4:00	116:00 h
10	18.07.2009	Documentation	Documentation	13.00 h	15.00 h		2:00	114:00 h
10	18.07.2009	Documentation	Documentation	23.00 h	01.30 h		2:30	111:30 h
10	19.07.2009	Documentation	Documentation	13.00 h	15.00 h		2:00	109:30 h
11	20.07.2009	Documentation	Documentation	08.00 h	19.30 h	03.00 h	8:30	104:00 h
11	20.07.2009	Développement	WebServices, SSO Moodle, Suggestions	22.00 h	02.30 h		4:30	99:30 h
11	21.07.2009	Documentation	Documentation	07.30 h	10.00 h		2:30	97:00 h
11	21.07.2009	Divers	Séance hebdomadaire	10.00 h	12.00 h		2:00	95:00 h
11	21.07.2009	Documentation	Documentation	13.00 h	15.00 h		2:00	93:00 h
11	21.07.2009	Divers	Rédaction PV	15.00 h	15.30 h		0:30	92:30 h
11	21.07.2009	Documentation	Documentation	16.00 h	02.00 h	02.00 h	8:00	86:30 h
11	22.07.2009	Documentation	Documentation	08.00 h	19.30 h	01.30 h	10:00	78:00 h
11	23.07.2009	Documentation	Documentation	10.00 h	20.30 h	01.30 h	9:00	70:30 h
11	24.07.2009	Documentation	Documentation	08.00 h	23.00 h	03.00 h	12:00	61:30 h
11	25.07.2009	Flash	Prise en main Flash	10.00 h	13.30 h		3:30	58:00 h
11	25.07.2009	Flash	Correction Bugs Flash	18.00 h	22.00 h		4:00	54:00 h
11	26.07.2009	Flash	Gestion MultiTypes + Corrections de bugs, ajouts de boutons, orthographe	00.00 h	04.00 h		4:00	50:00 h
11	26.07.2009	Flash	Corrections de bug	12.00 h	23.00 h	03.00 h	8:00	45:00 h
12	27.07.2009	Flash	Inviter des amis + adaptations	09.00 h	19.00 h	01.30 h	8:30	38:00 h
12	27.07.2009	Flash	Correction de bugs	20.00 h	04.00 h		8:00	30:00 h
12	28.07.2009	Flash	Correction de bugs	08.00 h	19.00 h	01.30 h	9:30	22:00 h
12	29.07.2009	Flash	Correction de bugs	08.00 h	19.00 h	01.30 h	9:30	14:00 h
12	30.07.2009	Documentation	Documentation	08.00 h	19.00 h	01.30 h	9:30	6:00 h
12	31.07.2009	Documentation	Documentation	08.00 h	18.00 h	01.30 h	8:30	-1:00 h
Total						40:00	401:00	-1:00 h



9.5 CHOIX D'ARCHITECTURE FLASH - FACEBOOK

9.5.1 ARCHITECTURES POSSIBLES ENTRE FLASH ET FACEBOOK

Dans ce projet de "Drill on Facebook", le principe est de créer un jeu flash pouvant être utilisé depuis une plateforme Moodle ou Facebook. Nous allons, dans ce document, nous pencher sur les différentes architectures possibles et imaginables.

Il existe premièrement deux possibilités fondamentalement différentes au niveau du choix de l'intégration de notre application par Facebook. Ce choix sera à faire lors de la création de l'application sur *Facebook*.

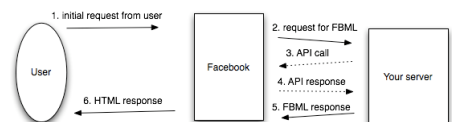
FBML⁶ ou iFrame

Nous allons maintenant passer en revue les différences entre ces solutions afin de pouvoir les comparer, et choisir la solution adaptée à ce projet.

9.5.2 DIFFÉRENCES ENTRE FBML ET IFRAME

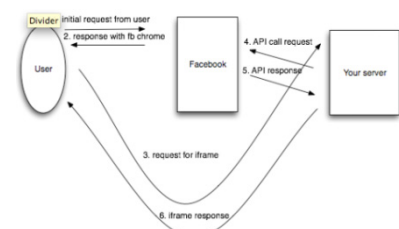
FBML :

- Permet de créer rapidement une application en partant de zéro
- Potentiellement plus rapide lors du chargement des pages
- Permet un accès facilité à beaucoup d'éléments Facebook
- Garde une certaine clarté des URL
- Possède un mécanisme d'authentification sensible



iFrame :

- Plus facile d'utilisation lorsque l'on a un site ou une application existante
- Potentiellement plus rapide sur la durée de l'application
- Permet d'utiliser du JavaScript, HTML et CSS
- Plus rapide lorsque que l'on utilise beaucoup d'AJAX car par besoin de passer par le proxy de Facebook
- Débogage plus aisé de l'HTML/JavaScript qu'avec les outils fournis pour le FBML et FBJS⁷



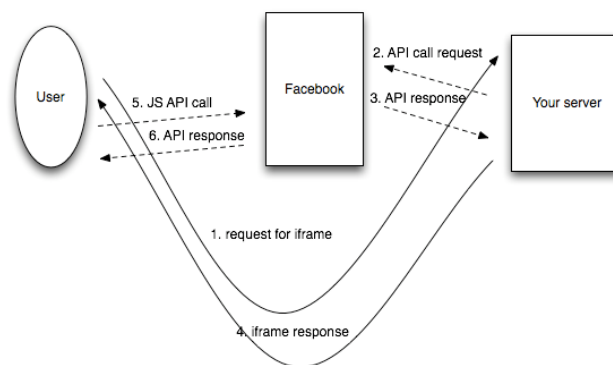
⁶ FaceBook Markup Language : Langage se rapprochant de l'HTML. Il est constitué de certains tags HTML avec en plus un set de tags propriétaires. Par exemple, le tag *fb:header* correspond au tag HTML *header*, alors que le tag *fb:profile-pic* est propriétaire au FBML.



9.5.3 COMPARAISON ENTRE FBML ET IFRAME

9.5.3.1 VITESSE

Nous l'avons vu plus haut, une solution basée sur du FBML est potentiellement plus rapide. Cela est toutefois à relativiser, car depuis la mise en service du XFBML⁸, les appels à l'API ne sont plus nécessaires depuis le serveur de l'application car ils vont se faire par un JavaScript directement depuis le navigateur vers Facebook. Tout ceci en utilisant le cache du navigateur, ce qui va donner la possibilité pour certaines pages de ne même pas faire un seul appel à l'API. Nous nous rapprochons donc d'un délai de réponse d'un site standard avec, en sus, les données Facebook qui vont s'y ajouter une fois la page chargée.



⁷ FaceBook JavaScript : langage propriétaire permettant au développeur d'utiliser des fonctions JavaScript tout en garantissant une confidentialité des données personnelles des utilisateurs.

⁸ eXternal FBML : utilisation de code FBML dans une iFrame.



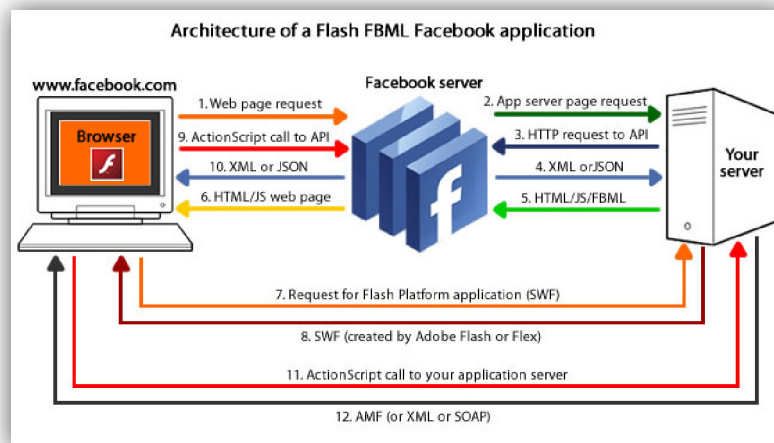
9.5.4 SOLUTIONS POSSIBLES

Notre jeu étant en flash, voici les deux solutions d'architectures possibles :

9.5.4.1 FLASH FBML FACEBOOK APPLICATION

Avec l'utilisation du FBML, voici les étapes effectuées lors du chargement de la page :

1. Requête de la page web
2. Facebook demande la page au serveur de l'application
3. Requête http à l'API
4. Réponse de l'API
5. Le serveur de l'application renvoie la page à facebook
6. Retour de la page de facebook au navigateur
7. Requête du flash
8. Retour du flash
9. Appels à l'API par ActionScript
10. Retour de l'API
11. Appels au serveur de l'application par ActionScript
12. Retour du serveur de l'application



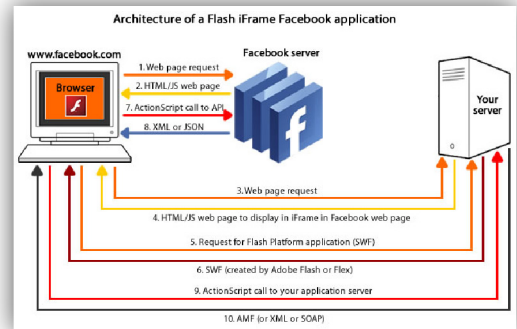
Un désavantage de cette solution est la complication du code, car la couche logique et la couche de présentation sont situées dans plusieurs endroits différents (dans le fichier SWF, la page web du serveur d'application et la page retournée à Facebook)



9.5.4.2 FLASH IFRAME FACEBOOK APPLICATION

Avec l'utilisation d'une iFrame, voici les étapes effectuées lors du chargement de la page :

1. Requête de la page web
2. Retour de la page de facebook au navigateur
3. Demande de l'iframe au serveur de l'application
4. Retour de l'iframe du serveur de l'application
5. Requête du flash
6. Retour du flash
7. Appels à l'API par ActionScript
8. Retour de l'API
9. Appels au serveur de l'application par ActionScript
10. Retour du serveur de l'application



9.6 CHOIX DE LA SOLUTION

Comparatif des deux solutions :

Description	iFrame	FBML
Contrôle de l'intégration du Flash	Oui	Non
Détection de la version du lecteur Flash	Oui	Non
Accès aux scripts	Oui	Limité
Mode plein écran	Oui	Non
Intégration de FBML	Oui	Oui
Appels API depuis le SFW	Oui	Oui
Contrôle de l'historique des URL	Oui	Oui

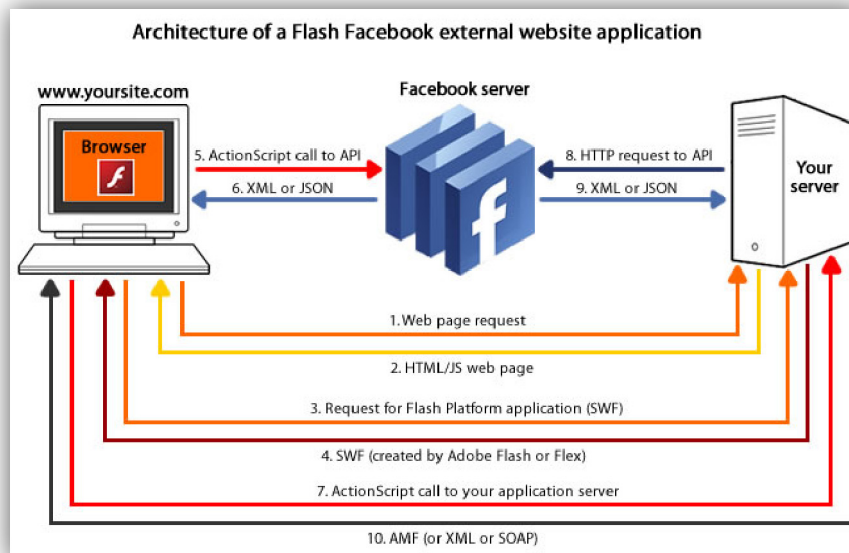
Selon ce tableau comparatif, ainsi qu'au vu des précisions vues plus haut, le choix s'est naturellement porté vers la solution :

Flash iFrame Facebook application



9.7 EXTERNAL FLASH FACEBOOK WEB APPLICATION

En ce qui concerne l'utilisation du jeu depuis Moodle, le fichier swf sera hébergé sur le serveur HESSODriller et fera appel, lorsqu'il y a besoin, au serveur Facebook selon le schéma suivant :



Les étapes 5, 6 et 8, 9 ne seront utilisées que dans le cas où des interactions avec la plateforme Facebook seront nécessaires (publication du résultat, tableau comparatif, etc.).



9.8 EXEMPLE PHPDOCUMENTOR

BusinessClasses

[class tree: BusinessClasses] [index: BusinessClasses] [all elements]

Todo List

Packages:
HES-SO_Driller
BusinessClasses
RemoteClasses
Tools

Files:
Answer.php
Comment.php
Course.php
Course_Question_Item.php
Game.php
Level.php
Login.php
Member.php
Member_Course_Access.php
Member_Role.php
Question.php
Question_State.php
Question_Type.php
Suggestion.php
Timelimit.php

Classes:
Answer
Comment
Course
Course_Question_Item
Game
Level
Login
Member
Member_Course_Access
Member_Role
Question
Question_State
Question_Type
Suggestion
Timelimit

Class: Member

Source Location: /classes/Member.php

Class Overview

Author(s):

- Andre Lambelet

Variables

- \$facebookUid
- \$FB_PREFIX
- \$globalHighscore
- \$lastsession
- \$MYSQL_TABLE
- \$role
- \$role_id
- \$username

Methods

- Member
- add
- calculatePoints
- delete
- getAuthenticated
- getFacebookMember
- getObjects
- getSingleMember
- loadFacebookProfile
- loadMoodleProfile
- loadProfile
- loadRole
- registerFacebookUser
- registerLastSession
- show
- update
- __toString

Class Details

[line 12]

Tags:
author: Andre Lambelet
since: 01.07.2009

[Top]

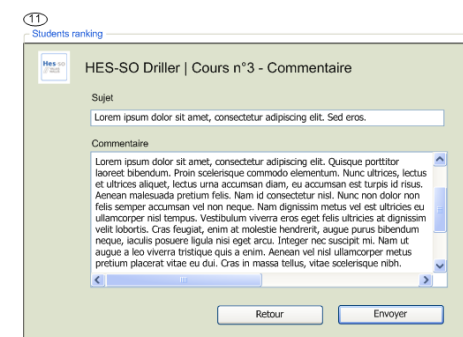
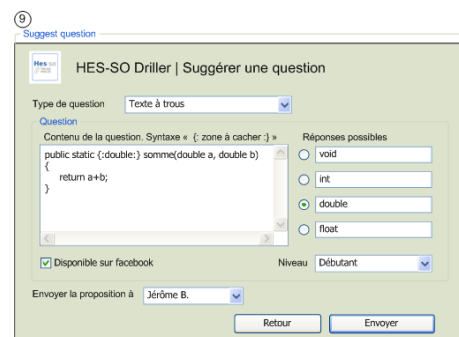
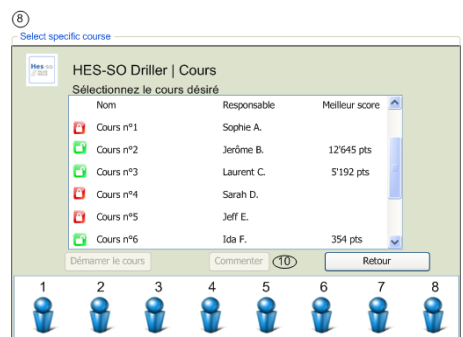
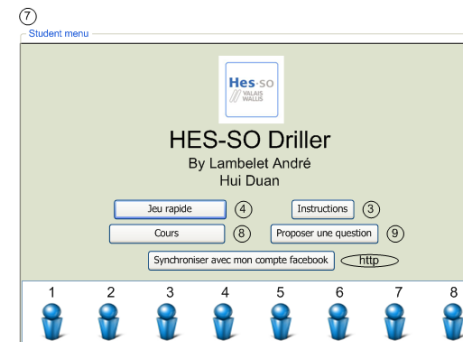
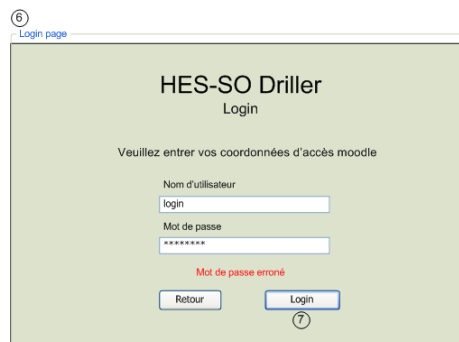
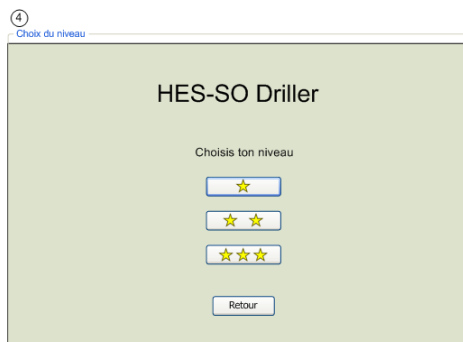
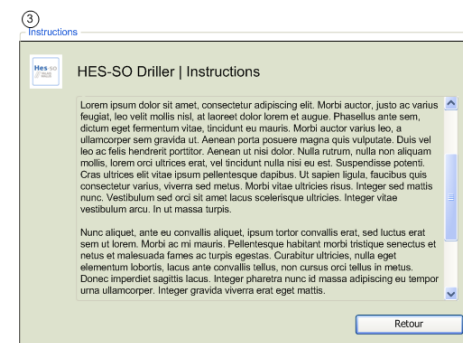
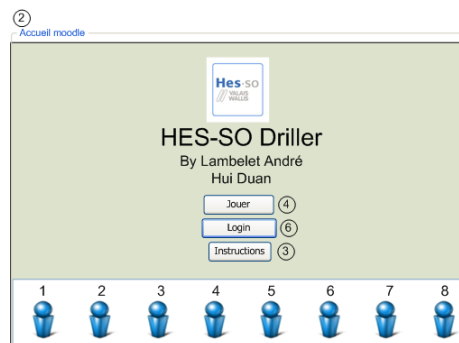
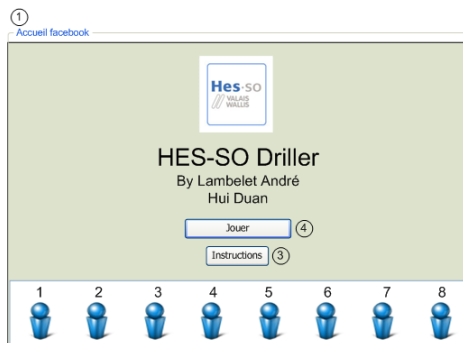
Class Variables

static \$FB_PREFIX = "facebook_"

[line 22]

Lambelet André - 2009

9.9 PROTOTYPES ÉCRANS



9.10 METHODES DU WEBSERVICE HESSODRILLER

Classe	Méthode	Retour	Commentaire
RemoteAnswer	set ([gameId], username, questionIds, time, true)	Boolean	questionIds, time, true = listes
RemoteComment	write(username, courseId, title, text)	Boolean	
RemoteCourse	hasAccess(username, courseid, [password])	Boolean	
RemoteCourse	getAllCourses(username)	Course[]	
RemoteCourse	getLevels()	Array(levelId=>description)	
RemoteGame	getGames([courseid])	Game[]	Jeux actifs avec questions
RemoteMember	getAuthenticated(username, password)	Member	
RemoteMember	getFacebookRanking(userid, fbFriendsUid)	Member[]	
RemoteMember	getFacebookUser(facebookUid)	Member	
RemoteMember	getMoodleRanking(userid, cours)	Member[]	
RemoteMember	getMoodleUser(username, [courseId])	Member[]	
RemoteMember	getTeachers()	Member[]	
RemoteQuestion	getAllQuestions([level])	Question[]	Toutes les questions si level null
RemoteQuestion	getGameQuestions([gameId])	Question[]	
RemoteQuestion	writeProposition(userid, typeId, level, xml, assignedTo, titre)	1 = ok, String = error	xml = urlencode(xml)
RemoteQuestion_Type	getTypes	Question_Type[]	



10 TABLE DES ILLUSTRATIONS

Image 1 - HES-SO Driller	1
Image 2 - Architecture physique	9
Image 3 - Architecture 5-tiers	10
Image 4 - Type de question « Texte à trous »	19
Image 5 - Type de question « Complétez le code »	20
Image 6 - Type de question « Réorganisez les lignes de code »	20
Image 7 - Logo amfphp	22
Image 8 - Logo ezSQL	22
Image 9 - Logo Facebook	23
Image 10 - Logo FacebookBridge	23
Image 11 - Logo Smarty	24
Image 12 - Logo TinyMCE	24
Image 13 - Ajout d'états supplémentaires	29
Image 14 - Type "trouvez l'erreur"	30
Image 15 - Bouton "Inviter des amis"	32
Image 16 - Formulaire d'invitation d'amis	32
Image 17 - Paramètres de configuration Moodle	44
Image 18 - Menu administration	47
Image 19 - Administration des questions	47
Image 20 - Gestion des cours	48
Image 21 - Edition d'un cours	48
Image 22 - Gestion d'un jeu	49
Image 23 - Ajout de questions à un jeu	49
Image 24 - Gestion des commentaires	49
Image 25 - Modifier un type de question	50
Image 26 - Gestion des droits	50



11 TABLE DES TABLEAUX

Tableau 1 - Droits d'accès	25
Tableau 2 - Configuration Facebook.....	45
Tableau 3 - Configuration des variables	46
Tableau 4 - Graphique des heures / catégories par jour.....	58
Tableau 5 - Graphique des proportions par catégorie.....	59

12 TABLE DES SCHÉMAS

Schéma 1 - Modèle physique de la base de données	14
Schéma 3 - BDD Cours et Jeux.....	15
Schéma 2 - BDD Membres.....	15
Schéma 4 - BDD Questions	16
Schéma 5 - Accès aux cours et commentaires.....	17
Schéma 6 - Navigation dans l'application.....	18
Schéma 7 - Déroulement du jeu.....	18
Schéma 8 - Diagramme de séquence de l'authentification	26
Schéma 9 - Types de questions dynamiques	28
Schéma 10 - Flash (Moodle) Obtention des informations des utilisateurs	31
Schéma 11 - Flash (Facebook) Obtention des informations des utilisateurs	31
Schéma 12 - Génération du formulaire.....	33
Schéma 13 - Génération du formulaire de modification.....	36
Schéma 14 - Structure des fichiers.....	53
Schéma 15 - Planification initiale.....	56
Schéma 16 - Planification finale.....	57



13 TABLE DES SOURCES

Source 1 - Sécurité de l'identification venant de Moodle.....	27
Source 2 - Définition de l'élément <i>donnee</i> (Template.xml).....	34
Source 3 - Configuration du type "question à 4 réponses" (XML).....	34
Source 4 - Schéma de validation du type "question à 4 réponses" (XSD).....	35
Source 5 - Validation de la question	35
Source 6 - Génération du Template.xml par PHP	36
Source 7 - HTML2XML.....	38
Source 8 - Page d'invitation d'amis	39
Source 9 - Algorithme de SSO modifié.....	41
Source 10 - Méthode getMember() originale	42
Source 11 - Méthode getMember() modifiée	42
Source 12 - Exemple phpDoc, classe Member	54



14 TABLE DES MATIÈRES

1	Remerciements	4
2	Glossaire	5
3	Présentation du projet.....	6
3.1	Contexte.....	6
3.2	Délivrables	6
3.3	Cahier des charges	7
4	Description de l'application.....	8
4.1	Introduction	8
4.2	Architecture physique	8
4.2.1	Acteurs principaux	8
4.2.2	Serveur applicatif.....	8
4.2.3	Serveur Moodle.....	8
4.2.4	Vue globale	9
4.3	Architecture applicative.....	10
4.3.1	Couche Physique	11
4.3.2	Couche Données	11
4.3.3	Couche Métier	11
4.3.4	Couche Application.....	11
4.3.5	Couche Présentation.....	12
4.4	Informations techniques.....	13
4.4.1	Introduction	13
4.4.2	Choix de l'architecture logicielle	13
4.4.2.1	Intégration de flash à facebook	13
4.4.3	Modélisation.....	13
4.4.3.1	Modèle complet.....	14
4.4.3.2	Description.....	15
	Membres.....	15
	Cours.....	15
	Questions	16
	Accès aux Cours et commentaires	17
4.4.4	Résultats de l'analyse	18
4.4.4.1	Navigation dans le jeu.....	18



4.4.4.2	Types de questions	19
	Question à 4 réponses.....	19
	Complétez le code.....	20
	Trouvez l'erreur.....	20
	Réorganisez les lignes de code	20
4.4.4.3	Contenu du Méta-cours	21
4.4.4.4	Prototypes écrans	21
4.4.4.5	Méthodes du Webservice	21
4.4.4.6	Calcul du score	21
4.4.5	Classes externes.....	22
4.4.5.1	amfphp	22
4.4.5.2	Ezsql.....	22
4.4.5.3	API Facebook.....	23
4.4.5.4	FacebookBridge.....	23
4.4.5.5	Fileupload	23
4.4.5.6	HTML2XML	24
4.4.5.7	Smarty	24
4.4.5.8	TinyMCE.....	24
4.4.5.9	XMLForm.....	24
4.4.5.10	xslt-php4-to-php5	24
4.5	Spécificités de l'implémentation	25
4.5.1	Niveaux d'accès	25
4.5.2	Web Service moodle	26
	4.5.2.1 Authentification	26
	4.5.2.2 Récupération du profil	27
4.5.3	Types de questions dynamiques.....	28
4.5.4	Gestion de l'état des questions.....	29
4.5.5	Flash.....	30
4.6	Analyse et explication des problèmes rencontrés.....	33
4.6.1	Types de question dynamiques	33
	4.6.1.1 Problématique	33
	4.6.1.2 Résolution	33
4.6.2	Sessions Facebook.....	37
	4.6.2.1 Problématique	37
	4.6.2.2 Résolution	37
4.6.3	TinyMCE & HTML2XML	38
	4.6.3.1 Problématique	38



4.6.3.2	Résolution	38
4.6.4	Formulaire d'invitation d'amis.....	39
4.6.4.1	Problématique	39
4.6.4.2	Résolution	39
4.7	Améliorations proposées	40
4.7.1	Interactions avec Facebook.....	40
4.7.2	Statistiques des réponses des étudiants.....	40
4.7.3	Sécurité.....	41
4.7.4	Gestion des droits automatiques	42
4.8	Documentation.....	43
4.8.1	Introduction.....	43
4.8.2	Déploiement	43
4.8.2.1	Spécifications minimales du serveur.....	43
4.8.2.2	Installation	44
	HESSODriller.....	44
	WebService Moodle.....	44
4.8.2.3	Configuration	44
	Moodle.....	44
	Facebook.....	45
	HESSODriller.....	46
4.8.3	Administration.....	47
4.8.3.1	Questions	47
4.8.3.2	Cours	48
4.8.3.3	Jeu	49
4.8.3.4	Commentaires	49
4.8.3.5	Types.....	50
4.8.3.6	Droits	50
4.8.4	Utilisation.....	51
4.8.5	Développement.....	53
4.8.5.1	Structure.....	53
4.8.5.2	Documentation du code.....	54
5	Gestion de projet.....	55
5.1	Déroulement	55
5.2	Répartition des tâches	56
5.3	Planification	56
5.4	Suivi hebdomadaire.....	57



5.5	Sauvegardes.....	58
5.6	Bilan des heures effectuées	58
5.7	Travail restant.....	59
5.8	Bilan final	59
6	Satisfaction personnelle.....	60
7	Déclaration sur l'honneur.....	61
8	Bibliographie	62
8.1	Général	62
8.2	Facebook.....	62
8.3	Flash.....	62
8.4	JavaScript	63
8.5	PHP	63
8.6	XML	63
8.7	Images.....	63
9	Annexes.....	64
9.1	Cahier des charges	64
9.2	Planification initiale	69
9.3	Procès-verbaux	70
9.4	Heures effectuées	81
9.5	Choix d'Architecture Flash - Facebook	83
9.6	Choix de la solution.....	86
9.7	External Flash facebook web application	87
9.8	Exemple phpDocumentor.....	88
9.9	Prototypes écrans	89
9.10	Methodes du Webservice HESSODriller.....	90
10	Table des illustrations.....	91
11	Table des tableaux	92
12	Table des schémas.....	92
13	Table des sources	93
14	Table des matières	94

